An Afterstates Reinforcement Learning Approach to Optimize Admission Control in Mobile Cellular Networks

Jose Manuel Gimenez-Guzman, Jorge Martinez-Bauset and Vicent Pla

Departamento de Comunicaciones, Universidad Politecnica de Valencia, UPV
ETSIT Camino de Vera s/n, 46022, Valencia, Spain
{jogiguz}@doctor.upv.es,{jmartinez,vpla}@dcom.upv.es

Abstract. We deploy a novel Reinforcement Learning optimization technique based on afterstates learning to determine the gain that can be achieved by incorporating movement prediction information in the session admission control process in mobile cellular networks. The novel technique is able to find better solutions and with less dispersion. The gain is obtained by evaluating the performance of optimal policies achieved with and without the predictive information, while taking into account possible prediction errors. The prediction agent is able to determine the handover instants both stochastically and deterministically. Numerical results show significant performance gains when the predictive information is used in the admission process, and that higher gains are obtained when deterministic handover instants can be determined.

1 Introduction

Session Admission Control (SAC) is a key traffic management mechanism in mobile cellular networks to provide QoS guarantees. Terminal mobility makes it very difficult to guarantee that the resources available at the time of session setup will be available in the cells visited during the session lifetime, unless a SAC policy is exerted. The design of the SAC system must take into account not only packet level issues (like delay, jitter or losses) but also session level issues (like loss probabilities of both session setup and handover requests). This paper explores the second type of issues from a novel optimization approach that exploits the availability of movement prediction information. To the best of our knowledge, applying optimization techniques to this type of problem has not been sufficiently explored. The results provided define theoretical limits for the gains that can be expected if handover prediction is used, which could not be established by deploying heuristic SAC approaches.

In systems that do not have predictive information available, both heuristic and optimization approaches have been proposed to improve the performance of the SAC at the session level. Optimization approaches not using predictive information have been studied in [1–4]. In systems that have predictive information available, most of the proposed approaches to improve performance are heuristic, see for example [5, 6] and references therein.

Our work has been motivated in part by the study in [5]. Briefly, the authors propose a sophisticated movement prediction system and a SAC scheme that taking advantage of movement prediction information is able to improve system performance. One of the novelties of the proposal is that the SAC scheme takes into consideration not only incoming handovers to a cell but also the outgoing ones. The authors justify it by arguing that considering only the incoming ones would led to reserve more resources than required, given that during the time elapsed since the incoming handover is predicted and resources are reserved until it effectively occurs, outgoing handovers might have provided additional free resources, making the reservation unnecessary.

In this paper we explore a novel Reinforcement Learning (RL) optimization technique based on afterstates learning, which was suggested in [7]. RL is a simulation based optimization technique in which an agent learns an optimal policy by interacting with an environment which rewards the agent for each executed action. We will show that when comparing afterstates learning with conventional learning, the former is able to find better solutions (policies) and with more precision (less dispersion).

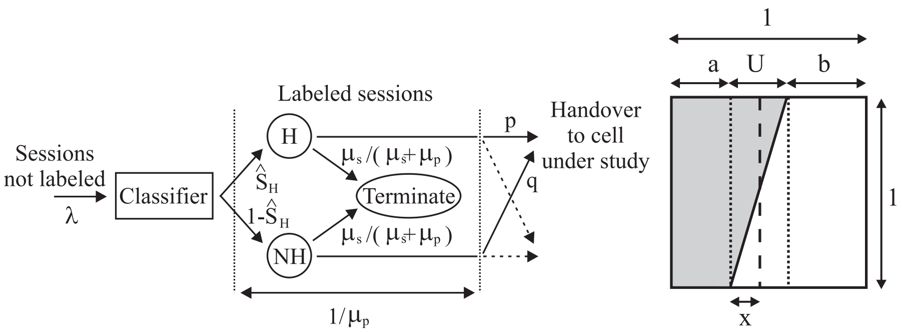
We do a comparative performance evaluation of different scenarios that differ on the type of predictive information that is provided to the SAC optimization process, like only incoming, only outgoing and both types of handover predictions together. We also evaluate the impact that predicting the future handover instants either stochastically or deterministically have on the system performance.

The rest of the paper is structured as follows. In Section 2 we describe the models of the system and of the two prediction agents deployed. The optimization approaches are presented in Section 3. A numerical evaluation comparing the performance obtained when using different types of information and when handovers instants are stochastically or deterministically predicted is provided in Section 4. This later Section also includes a comparison of the performance of the two reinforcement learning approaches, i.e. afterstates and conventional learning. Finally, a summary of the paper and some concluding remarks are given in Section 5.

2 Model Description

We consider a single cell system and its neighborhood, where the cell has a total of C resource units, being the physical meaning of a unit of resources dependent on the specific technological implementation of the radio interface. Only one service is offered but new and handover session arrivals are distinguished, making a total of two arrival types.

For mathematical tractability we make the common assumptions. New and handover sessions arrive according to a Poisson process with rates λ_n and λ_h respectively. The duration of a session and the cell residence time are exponentially distributed with rates μ_s and μ_r respectively, hence the resource holding time in a cell is also exponentially distributed with rate $\mu = \mu_s + \mu_r$. Without



(a) Basic operation of the IPA

(b) Basic parameters of the classifier

Fig. 1. IPA and classifier models.

loss of generality, we will assume that each session consumes one unit of resource and that only one session is active per mobile terminal (MT).

We used a model of the prediction agent, given that the focus of our study was not the design of it.

2.1 Prediction Agent for Incoming Handovers

An active MT entering the cell neighborhood is labeled by the prediction agent for incoming handovers (IPA) as "probably producing a handover" (H) or the opposite (NH), according to some of its characteristics (position, trajectory, velocity, historic profile,...) and/or some other information (road map, hour of the day,...). After an exponentially distributed time, the actual destiny of the MT becomes definitive and either a handover into the cell occurs or not (for instance because the session ends or the MT moves to another cell) as shown in Fig. 1(a). The SAC system is aware of the number of MTs labeled as H at any time.

The model of the classifier is shown in Fig. 1(b) where the square (with a surface equal to one) represents the population of active MTs to be classified. The shaded area represents the fraction of MTs (S_H) that will ultimately move into the cell, while the white area represents the rest of active MTs. Notice that part of the MTs that will move into the cell can finish their active sessions before doing so. The classifier sets a threshold (represented by a vertical dashed line) to discriminate between those MTs that will likely produce a handover and those that will not. The fraction of MTs falling on the left side of the threshold ($\hat{S}_H$) are labeled as H and those on the right side as NH. There exists an uncertainty zone, of width U, which accounts for classification errors: the white area on the left of the threshold ($\hat{S}^e_H$) and the shaded area on the right of the threshold ($\hat{S}^e_{NH}$). The parameter x represents the relative position of the classifier threshold within the uncertainty zone. Although for simplicity we use a linear model for the uncertainty zone it would be rather straightforward to consider a different model.

As shown in Fig. 1(a), the model of the IPA is characterized by three parameters: the average sojourn time of the MT in the predicted stage μ_p^{-1}, the probability p of producing a handover if labeled as H and the probability q of producing a handover if labeled as NH. Note that $1-p$ and q model the false-positive and non-detection probabilities respectively and in general $q \neq 1-p$.

From Fig. 1(b) it follows that

$$1-p = \frac{\hat{S}_H^e}{\hat{S}_H} = \frac{x^2}{2U(a+x)} \ ; \qquad q = \frac{\hat{S}_{NH}^e}{1-\hat{S}_H} = \frac{(U-x)^2}{2U(1-a-x)}$$

Parameters a and b can be expressed in terms of S_H and U, being $a = S_H - U/2$ and $b = 1 - S_H - U/2$. Then

$$1-p = \frac{x^2}{(U(2S_H - U + 2x))}; \quad q = \frac{(U-x)^2}{(U(2-2S_H+U-2x))} \tag{1}$$

Referring to Fig. 1(a), the value of the session rate entering the classifier λ is chosen so that the system is in statistical equilibrium, i.e. the rate at which handover sessions enter a cell (λ_h^{in}) is equal to the rate at which handover sessions exit the cell (λ_h^{out}). It is clear that

$$\lambda_h^{in} = \lambda S_H \frac{\mu_p}{\mu_p + \mu_s} \ ; \qquad \lambda_h^{out} = \frac{\mu_r}{\mu_r + \mu_s}[(1-P_n)\lambda_n + (1-P_h)\lambda_h^{in}]$$

where P_n (P_h) is the blocking probability of new (handover) requests.

Making $\lambda_h^{in} = \lambda_h^{out}$, substituting P_h by $P_h = (\mu_s/\mu_r) \cdot [P_{ft}/(1-P_{ft})]$, where P_{ft} is the probability of forced termination of a successfully initiated session, and after some algebra we get

$$\lambda = (1-P_n)(1-P_{ft})\lambda_n(\mu_r/\mu_s + \mu_r/\mu_p)(1/S_H) \tag{2}$$

2.2 Prediction Agent for Outgoing Handovers

The model of the prediction agent for outgoing handovers (OPA) is shown in Fig. 2. The OPA labels active sessions in the cell as H if they will produce a handover or as NH otherwise. The classification is performed for both handover sessions that enter the cell and new sessions that initiate in the cell, and are carried out by a classifier which model is the same as the one used in the IPA. The time elapsed since the session is labeled until the actual destiny of the MT becomes definitive is the cell residence time that, as defined, is exponentially distributed with rate μ_r. The fraction of sessions that effectively execute an outgoing handover is given by $S_H = \mu_r/(\mu_s + \mu_r)$. The OPA model is characterized by only two parameters $1-p$ and q, which meaning is the same as in the IPA model. Note that $1-p$ and q can be related to the classifier parameters by the expressions in (1).

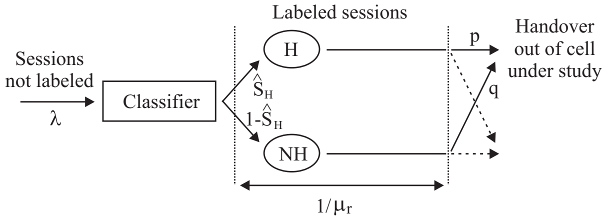


Fig. 2. Basic operation of the OPA.

In an earlier version of the IPA we were providing the optimization process only with state information of the neighboring cells but without any predictive information. We obtained that the gain was not significant, possibly because the information was not sufficiently specific. The authors in [8] reached the same conclusion but using a genetic algorithm to find near-optimal policies. As it will be described in Section 4, here we are considering a circular-shaped cell of radio r and a holed-disk-shaped neighborhood with inner (outer) radio $1.0r$ $(1.5r)$ and providing the optimization process with predictive information of this sufficiently close neighborhood. This produces a significant gain. For the design of the OPA we were faced with the same dilemma but in this case we decided not to use more specific information. Defining a holed-disk-shaped neighborhood with outer (inner) radio r $(< r)$ for the outgoing handovers and an exponentially distributed sojourn time in it, would had open the possibility of having terminals that could go in and out of this area, making the cell residence time not exponential. This would had increased the complexity and made the models with the IPA and with the OPA not comparable.

3 Optimizing the SAC Policy

We formulate the optimization problem as an infinite-horizon finite-state Markov decision process under the average cost criterion, which is more appropriate for the problem under study than other discounted cost approaches. When the system starts at state $\boldsymbol{x}$ and follows policy π then the average expected cost rate over time t, as $t \rightarrow \infty$, is denoted by $\gamma^{\pi}(\boldsymbol{x})$ and defined as: $\gamma^{\pi}(\boldsymbol{x}) = \lim_{t\rightarrow\infty} \frac{1}{t} E\left[w^{\pi}(\boldsymbol{x},t)\right]$, where $w^{\pi}(\boldsymbol{x},t)$ is a random variable that expresses the total cost incurred in the interval $[0,t]$. For the systems we are considering, it is not difficult to see that for every deterministic stationary policy the embedded Markov chain has a unichain transition probability matrix, and therefore the average expected cost rate does not vary with the initial state [9]. We call it the "cost" of the policy π, denote it by γ^{π} and consider the problem of finding the policy π^* that minimizes γ^{π}, which we name the optimal policy.

In our model the cost structure is chosen so that the average expected cost represents a weighted sum of the loss rates, i.e. $\gamma^{\pi} = \omega_n P_n \lambda_n + \omega_h P_h \lambda_h$, where ω_n (ω_h) is the cost incurred when the loss of a new (handover) request occurs and P_n $(P_h$ $)$ is the loss probability of new (handover) requests. In general, $\omega_n < \omega_h$

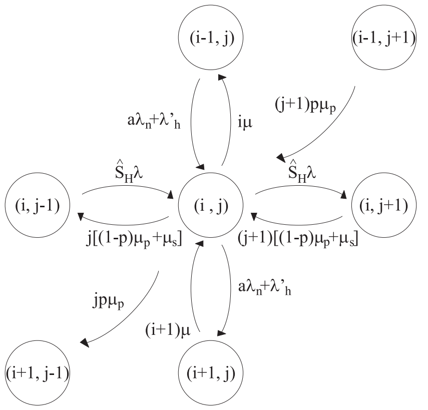


Fig. 3. State transition diagram.

since the loss of a handover request is less desirable than the loss of a new session setup request.

Two optimization approaches have been explored: a dynamic programming (DP) approach and an automatic learning approach based on the theory of Reinforcement Learning [7]. DP gives an exact solution and allows to evaluate the theoretical limits of incorporating handover prediction in the SAC system, whereas RL tackles more efficiently the curse of dimensionality and has the important advantage of being a model-free method, i.e. transition probabilities and average costs are not needed in advance. In both approaches handover sessions have priority over new sessions and they are accepted as long as resources are available.

3.1 Dynamic Programming

We apply DP to the scenario that only considers the incoming handovers, in which case the system state space is $S := \{\boldsymbol{x} = (i, j) : 0 \leq i \leq C;\ 0 \leq j \leq C_p\}$, where i is the number of active sessions in the cell, j is the number of MTs labeled as H in the cell neighborhood and C_p is the maximum number of MT that can be labeled as H at a given time. We use a large value for C_p so that it has no practical impact in our results. At each state $(i, j), i < C$, the set of possible actions is defined by $A := \{a : a = 0, 1\}$, being $a = 0$ the action that rejects an incoming new session and $a = 1$ the action that accepts an incoming new session. The system can be described as a continuous-time Markov chain which state transition diagram is shown in Fig. 3, where $\lambda'_h = q\lambda(1 - \hat{S}_H)\mu_p/(\mu_p + \mu_s)$ denotes the average arrival rate of unpredicted handovers. It is converted to a *discrete time Markov chain* (DTMC) by applying uniformization. It can be shown that $\Gamma = C_p(\mu_p + \mu_s) + C(\mu_r + \mu_s) + \lambda + \lambda_n$ is an uniform upper-bound for the outgoing rate of all the states, being λ the input rate to the classifier. If $r_{\boldsymbol{xy}}(a)$ denotes the transition rate from state $\boldsymbol{x}$ to state $\boldsymbol{y}$ when action a is taken at state $\boldsymbol{x}$, then the transition probabilities of the resulting DTMC are given by $p_{\boldsymbol{xy}}(a) = r_{\boldsymbol{xy}}(a)/\mathit{\Gamma}$ $(\boldsymbol{y} \neq \boldsymbol{x})$ and $p_{\boldsymbol{xx}}(a) = 1 - \sum_{\boldsymbol{y} \in S} p_{\boldsymbol{xy}}(a)$. We define the incurred cost rate at state $\boldsymbol{x}$ when action a is selected by $c(\boldsymbol{x}, a)$, which can take any of the following values: 0 $(i < C,\ a = 1)$, $\omega_n \lambda_n$ $(i < C,\ a = 0)$ or $\omega_n \lambda_n + \omega_h(\lambda_h' + jp\mu_p)$ $(i = C,\ a = 0)$.

If we denote by $h(\boldsymbol{x})$ the relative cost rate of state $\boldsymbol{x}$ under policy π, then we can write

$$h(\boldsymbol{x}) = c(\boldsymbol{x}, \pi(\boldsymbol{x})) - \gamma^{\pi} + \sum_{\boldsymbol{y}} p_{\boldsymbol{xy}}(\pi(\boldsymbol{x}))h(\boldsymbol{y}) \qquad \forall \boldsymbol{x} \tag{3}$$

from which we can obtain the average cost and the relative costs $h(\boldsymbol{x})$ up to an undetermined constant. We arbitrarily set $h(0, 0) = 0$ and then solve the linear system of equations (3) to obtain γ^{π} and $h(\boldsymbol{x})$, $\forall \boldsymbol{x}$. Having obtained the average and relative costs under policy π, an improved policy π' can be calculated as

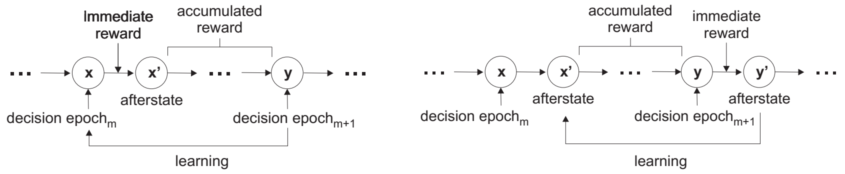
$$\pi'(\boldsymbol{x}) = \underset{a=0,1}{\arg\min} \left\{ c(\boldsymbol{x}, a) - \gamma^{\pi} + \sum_{\boldsymbol{y}} p_{\boldsymbol{xy}}(a)h(\boldsymbol{y}) \right\}$$

so that the following relation holds $\gamma^{\pi'} \leq \gamma^{\pi}$. Moreover, if the equality holds then $\pi' = \pi = \pi^*$, where π^* denotes the optimal policy, i.e. $\gamma^{\pi^*} \leq \gamma^{\pi}\ \forall \pi$.

We repeat iteratively the solution of system (3) and the policy improvement until we obtain a policy which does not change after improvement. This process is called *Policy Iteration* [9, Section 8.6] and it leads to the average optimal policy in a finite —and typically small— number of iterations. Note that although the number of iterations is typically small each iteration entails solving a linear system of the same size as the state space, and thus the overall computational complexity can be considerably high.

3.2 Reinforcement Learning

We formulate the optimization problem as an infinite-horizon finite-state semi-Markov decision process (SMDP) under the average cost criterion. Only arrival events are relevant to the optimization process because no actions are taken at session departures. Additionally, given that no decisions are taken for handover arrivals (they are always accepted if enough free resources are available), then the decision epochs correspond only to the time instants at which new session arrivals occur. The state space for the scenario that only considers the incoming handovers is the same as defined when deploying DP, i.e. $S := \{\boldsymbol{x} = (x_0, x_{in}) : x_0 \leq C; x_{in} \leq C_p\}$, where x_0 and x_{in} represent, respectively, the number of active sessions in the cell and the number of sessions labeled as H in the cell neighborhood. The state space for the scenario that only considers the outgoing handovers is defined as $S := \{\boldsymbol{x} = (x_0, x_{out}) : x_{out} \leq x_0 \leq C\}$, where x_{out} represents the number of sessions labeled as H in the cell. The state space for the scenario that considers both the incoming and outgoing handovers is defined as $S := \{\boldsymbol{x} = (x_0, x_{in}, x_{out}) : x_{out} \leq x_0 \leq C; x_{in} \leq C_p\}$. At each decision epoch the system has to select an action from the set $A := \{a : a = 0, 1\}$, being $a = 0$



(a) Conventional learning. (b) Afterstates learning.

Fig. 4. Reinforcement learning process.

the action that rejects an incoming new session and $a = 1$ the action that accepts an incoming new session.

The cost structure is defined as follows. At any decision epoch, the cost incurred by accepting a new session request is zero and by rejecting it is ω_n. Further accrual of cost occurs when the system has to reject handover requests between two decision epochs, incurring a cost of ω_h per rejection.

Intuitively, in systems as the one being considered, afterstates learning is based on the idea that what is relevant in the learning process is the state reached immediately after the action is taken. More specifically, all states at decision epochs in which the immediate actions taken drive the system to the same afterstate, would accumulate the same future cost if the same future actions are taken. The difference between conventional learning and afterstates learning is shown in Fig. 4.

In SMDPs actions occur at variable length time instants and therefore, state transition dynamics is specified not only by the state where an action was taken, but also by a parameter specifying the length of time since the action was taken. The Bellman optimality recurrence equations for a SMDP under the average cost criterion when learning is done at each decision epoch can be written as

$$h^*(\boldsymbol{x}) = \min_{a \in A_x} \{w(\boldsymbol{x}, a) - \gamma^* \tau(\boldsymbol{x}, a) + \sum_{\boldsymbol{y} \in S} p_{\boldsymbol{xy}}(a) \min_{a' \in A_{\boldsymbol{y}}} h^*(\boldsymbol{y}, a')\}$$

where $h^*(\boldsymbol{x}, a)$ is the average expected relative cost of taking the optimal action a in state $\boldsymbol{x}$ and then continuing indefinitely by choosing actions optimally, $w(\boldsymbol{x}, a)$ is the average cost of taking action a in state $\boldsymbol{x}$, $\tau(\boldsymbol{x}, a)$ is the average sojourn time in state $\boldsymbol{x}$ under action a (i.e. the average time between decision epochs) and $p_{\boldsymbol{xy}}(a)$ is the probability of moving from state $\boldsymbol{x}$ to state $\boldsymbol{y}$ under action $a = \pi(\boldsymbol{x})$.

We deploy a modified version of the SMART algorithm [10] which follows an afterstates learning process using a temporal difference method (TD(0)). The pseudo code of the proposed algorithm is shown in the box below. In systems where the number of states can be large, RL based on afterstates learning tackles more efficiently the curse of dimensionality.

SMART with afterstates

```
Initialize $h(\boldsymbol{x}), \forall \boldsymbol{x} \in S$ , arbitrarily (usually zeros)
Initialize $\gamma$ arbitrarily (usually zeros)
Initialize $N(\boldsymbol{x}) = 0$, $W_T = 0$ and $T_T = 0$
Repeat forever:
    We denote by $a$ the action taken in the current state $\boldsymbol{y}$, by $\boldsymbol{y'}_{reject}$
    ($\boldsymbol{y'}_{accept}$) the afterstate when the reject (accept) action is taken and
    by $\omega_{reject}$ the immediate cost when the request is rejected.
    Take action $a$:
        Exploration: random action
        Greedy: action selected from
            if $\big(h(\boldsymbol{y'}_{reject}) + \omega_{reject}\big) < h(\boldsymbol{y'}_{accept})$ then
                $a = reject$
            else
                $a = accept$
    $\alpha = 1/(1 + N(\boldsymbol{x'}))$
    being $\alpha$ de learning rate, $\boldsymbol{x'}$ the previous afterstate and $N(\boldsymbol{x'})$ the number of
    times the afterstate $\boldsymbol{x'}$ has been updated:
    $h(\boldsymbol{x'}) \leftarrow (1-\alpha)h(\boldsymbol{x'}) + \alpha\big[w_c(\boldsymbol{x'}, \boldsymbol{y}) + w(\boldsymbol{y}, a) + h(\boldsymbol{y'}) - \gamma\tau\big]$
    $N(\boldsymbol{x'}) \leftarrow N(\boldsymbol{x'}) + 1$
    being $w_c(\boldsymbol{x'}, \boldsymbol{y})$ the accrued cost when the system evolves from $\boldsymbol{x'}$ to $\boldsymbol{y}$, $w(\boldsymbol{y}, a)$
    the immediate cost of taking action $a$ in state $\boldsymbol{y}$ and $\tau$ the time elapsed
    between decision epochs $m$ and $m+1$ (see Fig. 4(b)).
    if $a$ is greedy:
        $W_T \leftarrow W_T + w_c(\boldsymbol{x'}, \boldsymbol{y}) + w(\boldsymbol{y}, a)$
        $T_T \leftarrow T_T + \tau$
        $\gamma \leftarrow W_T/T_T$
    $\boldsymbol{x'} \leftarrow \boldsymbol{y'}$
```

4 Numerical Evaluation

When introducing prediction, we evaluated the performance gain by the ratio $\gamma^{\pi}_{wp}/\gamma^{\pi}_{p}$, where γ^{π}_{p} (γ^{π}_{wp}) is the average expected cost rate of the optimal policy in a system with (without) prediction. We assume a circular-shaped cell of radio r and a holed-disk-shaped neighborhood with inner (outer) radio $1.0r$ ($1.5r$).

The values of the parameters that define the scenario are: $C = 10$ and $C_p = 60$, $\mu_r/\mu_s = 1$, $\mu_r/\mu_p = 0.5$, $\lambda_n = 2$, $\mu = \mu_s + \mu_r = 1$, $x = U/2$, $w_n = 1$, and $w_h = 20$. When deploying the IPA, $S_H = 0.4$. Note that in our numerical experiments the values of the arrival rates are chosen to achieve realistic operating values for $P_n(\approx 10^{-2})$ and $P_{ft}(\approx 10^{-3})$. For such values, we approximate (2) as $\lambda \approx 0.989\lambda_n(N_h + \mu_r/\mu_p)(1/S_H)$.

For the RL simulations, the ratio of arrival rates of new sessions to the cell neighborhood (ng) and to the cell (nc) is made equal to the ratio of their surfaces, $\lambda_{ng} = 1.25\lambda_{nc}$. The ratio of handover arrival rates to the cell neighborhood from the outside of the system (ho) and from the cell (hc) is made equal to ratio of their perimeters, $\lambda_{ho} = 1.5\lambda_{hc}$. Using the flow equilibrium property, we can

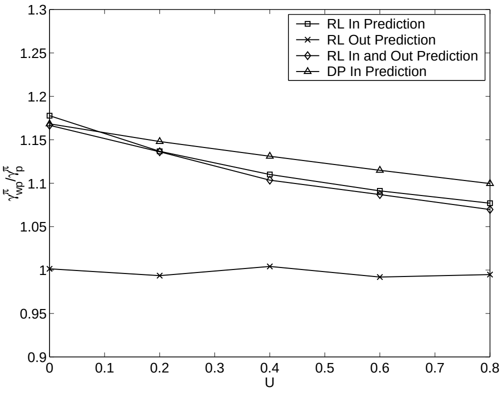


Fig. 5. Performance gain when using stochastic handover prediction.

write $\lambda_{hc} = (1 - P_n)(1 - P_{ft})(\mu_r/\mu_s)\lambda_{nc} \approx 0.989(\mu_r/\mu_s)\lambda_{nc}$. With regard to the RL algorithm, at the m^{th} decision epoch an exploratory action is taken with probability p_m, which is decayed to zero by using the following rule $p_m = p_0/(1 + u)$, where $u = m^2/(\varphi + m)$. We used $\varphi = 1.0 \cdot 10^{12}$ and $p_0 = 0.1$. The exploration of the state space is a common RL technique used to avoid being trapped at local minima.

4.1 Stochastic Prediction

The prediction agents described in Sections 2.1 and 2.2 predict the time instants at which handovers will occur only stochastically and Fig. 5 shows the gain for different values of the uncertainty U when deploying such agents. When using RL, for each value of U we run 10 simulations with different seeds and display the averages. As observed, using incoming handover prediction induces a gain and that gain decreases as the prediction uncertainty (U) increases. From Fig. 5 it is clear that the knowledge of the number of resources that will become available is not relevant for the determination of optimum SAC policies, being even independent of the degree of uncertainty. This counter-intuitive phenomenon could be explained as follows.

Lemma 1 *Let X and Y be two independent and exponentially distributed rv with means $1/\mu_x$ and $1/\mu_y$, and $f_{(X,Y)}(x,y)$ its joint pdf, where $f_{(X,Y)}(x,y) = f_X(x)f_Y(y)$. Then the pdf of X conditioned on $X < Y$, is given by*

$$f_X(x|X<Y) = \frac{\int_x^\infty f_{(X,Y)}(x,y)dy}{\int_0^\infty \int_x^\infty f_{(X,Y)}(x,y)dydx} = \frac{f_X(x)\int_x^\infty f_Y(y)dy}{\int_0^\infty \int_x^\infty f_X(x)f_Y(y)dydx}$$

$$= (\mu_x + \mu_y)e^{-(\mu_x+\mu_y)x}$$

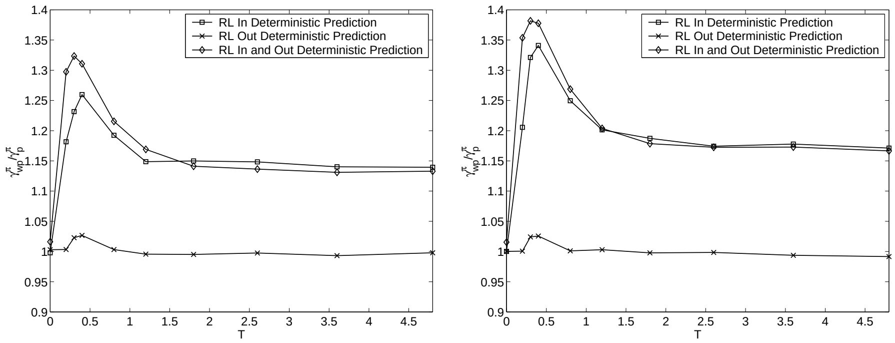
Now consider a perfect OPA, i.e. one with $p = 1$ and $q = 0$. Those sessions tagged as H will release the resources because they leave the cell —since we know this will happen before the session finishes— and hence, applying the result set in Lemma 1, the holding time of resources is exponentially distributed with mean $1/(\mu_r + \mu_s)$. Conversely, those sessions tagged as NH will release the resources because their sessions finish —since we know this will happen before the terminal leaves the cell— and hence the holding time of resources is exponentially distributed with mean $1/(\mu_r + \mu_s)$. Note that as the holding time of resources for H and NH sessions are identically distributed, having an imperfect OPA will not make any difference. On the other hand, if no out prediction is considered, an active session will release the resources because the session finishes or the terminal leaves the cell, whichever happens first, and therefore the holding time of resources is also exponentially distributed with mean $1/(\mu_r + \mu_s)$.

Therefore, if both the cell residence time and the session holding time are exponentially distributed, knowing whether a session will produce an outgoing handover or not does not provide, in theory, any helpful information to the SAC process. Additionally, the performance of the SAC should not be affected by the precision of the OPA.

4.2 Deterministic Prediction

In this section we evaluate the impact that more precise knowledge of the future handover time instants have on performance. Intuitively, it seems obvious that handovers taking place in a near future would be more relevant for the SAC process than those occurring in an undetermined far future. More precisely, in this is section both the IPA and OPA operate as before but the prediction will be made available to the admission controller, at most, T time units in advance the handover takes place. If the future handover is predicted less than T time units ahead of its occurrence the prediction is made available to the admission controller immediately, i.e. when the handover sessions enter the cell neighborhood or the cell and when new sessions are initiated. In that sense, the stochastic prediction can be seen as particular or limit case of the deterministic one when $T \to \infty$. A similar approach is used in [5], where authors predict the incoming and outgoing handovers that will take place in a time window of fixed size.

For the performance evaluation we use the scenarios, parameters and methodology described in Section 4.1, but now two uncertainty values are considered. In the first we set $U = 0.2$, which we consider it might be a practical value, while in the second, as a reference, we set it to $U = 0$. Figure 6 shows the variation of the gain for different values of T and U. As observed, there exists an optimum value for T, which is close to the mean time between call arrivals (λ^{-1}), although it might depend on other system parameters as well. As T goes beyond its optimum value, the gain decreases, probably because the temporal information becomes less significant for the SAC decision process. As expected, when $T \to \infty$ the gain is identical to the one in the stochastic prediction case. When T is lower than its optimum value the gain also decreases, probably because the



(a) U=0.2.

(b) U=0.

Fig. 6. Performance gain when using deterministic handover prediction.

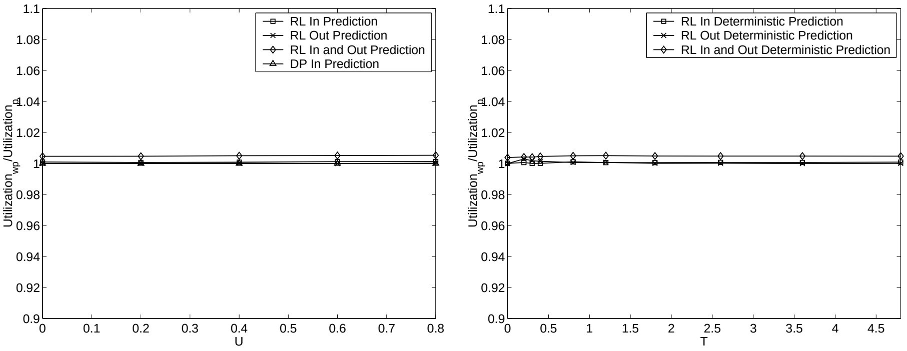
system has not enough time to react. When $T = 0$ the gain is null because there is no prediction at all.

Figure 6 shows again that the information provided by the OPA is in general not relevant for the optimization process, except for a small interval around its optimal value, which is slightly above unity. For values of T close to its optimum, the gain is higher when using incoming and outgoing prediction together than when using only incoming handover prediction, and it is significantly higher than when stochastic time prediction is used.

Finally it is worth noting that the main challenge in the design of efficient bandwidth reservation techniques for mobile cellular networks is to balance two conflicting requirements: reserving enough resources to achieve a low forced termination probability and keeping the resource utilization high by not blocking too many new setup requests. Figure 7, which shows the ratio of the system resources utilization when not using prediction and when using prediction (utilization$_{wp}$/utilization$_{p}$) for both stochastic and deterministic prediction, justifies the efficiency of our optimization approach.

4.3 Comparison of Learning Techniques

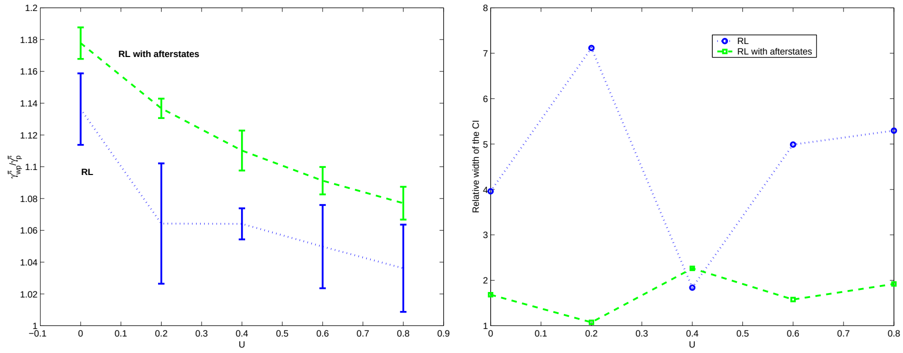
In this section we evaluate the performance of the afterstates learning process. Figure 8 and Figure 9 compares the mean, the confidence interval and the relative width of the confidence interval (the ratio of the width to the mean) when deploying conventional learning and afterstates learning. As observed, the solutions obtained when deploying afterstates learning are better (the gain $\gamma^{\pi}_{wp}/\gamma^{\pi}_{p}$ is higher) and more precise (the relative width of the confidence interval is smaller).



(a) Stochastic handover prediction.

(b) Deterministic handover prediction for U=0.2.

Fig. 7. Utilization gain when using handover prediction.



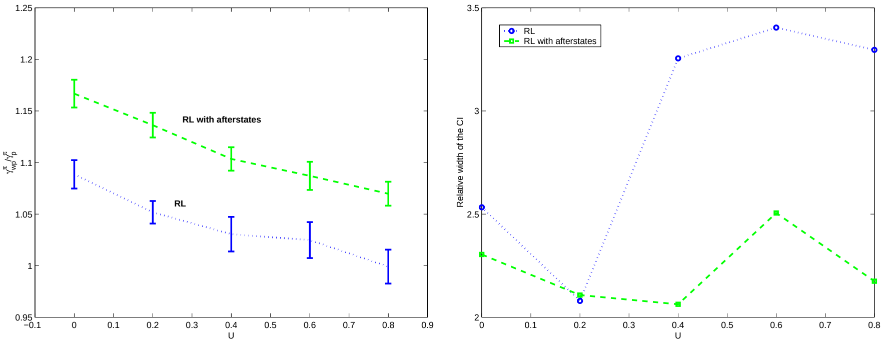
(a) Confidence interval of the gain when deploying input prediction.

(b) Relative width of the confidence interval when deploying input prediction.

Fig. 8. Comparison of the confidence intervals of the gain when deploying input prediction.

5 Conclusions

In this paper we evaluate the performance gain that can be expected when the SAC optimization process is provided with information related to incoming, outgoing and incoming and outgoing handovers together, in a mobile cellular network scenario. The prediction information is provided by two types of prediction agents that label active mobile terminals in the cell or its neighborhood which



(a) Confidence interval of the gain when deploying input and output prediction.

(b) Relative width of the confidence interval when deploying input and output prediction.

Fig. 9. Comparison of the confidence intervals of the gain when deploying input and output prediction.

will probably execute a handover. The prediction agents predict the future time instants at which handovers will occur either stochastically or deterministically.

The optimization problem is formulated as a Markov or semi-Markov decision process and two solving methods are used: dynamic programming and an afterstates based reinforcement learning approach. A general model of the prediction agents has been considered and as such it cannot be used neither to obtain results for specific systems nor to evaluate the added complexity of deploying a particular prediction method in operational systems. Nevertheless, the generality of the prediction model together with the optimization-based approach permit to obtain bounds for the gain of specific prediction schemes used in conjunction with SAC.

For the system model deployed, numerical results show that the information related to incoming handovers is more relevant than the one related to outgoing handovers. Additional performance gains can be obtained when more specific information is provided about the handover time instants, i.e. when their prediction is deterministic instead of stochastic. The gain obtained has been higher than 30% in the studied scenario even when the prediction uncertainty is 20%.

In a future work we will study the impact that a non-exponential resource holding time has on the performance of systems which deploy predictive information in the SAC process. As shown, when the resource holding time is exponential, deploying the OPA does not improve performance. We will also generalize the operation of the deterministic prediction agents by considering values for T independent for the IPA and for the OPA. Another aspect that deserves a closer study is the identification of the parameters that affect the optimum value of T and the study of its sensitivity.

Acknowledgments

This work has been supported by the Spanish Ministry of Education and Science (30% PGE, 70% FEDER) under projects TIC2003-08272 and TEC2004-06437-C05-01, and by the Universidad Politécnica de Valencia under "Programa de Incentivo a la Investigación".

References

1. R. Ramjee, R. Nagarajan, and D. Towsley, "On optimal call admission control in cellular networks," Wireless Networks Journal (WINET), vol. 3, no. 1, pp. 29–41, 1997.
2. N. Bartolini, "Handoff and optimal channel assignment in wireless networks," Mobile Networks and Applications (MONET), vol. 6, no. 6, pp. 511–524, 2001.
3. N. Bartolini and I. Chlamtac, "Call admission control in wireless multimedia networks," in Proceedings of IEEE PIMRC, 2002.
4. V. Pla and V. Casares-Giner, "Optimal admission control policies in multiservice cellular networks," in Proceedings of the International Network Optimization Conference (INOC), 2003, pp. 466–471.
5. W.-S. Soh and H. S. Kim, "Dynamic bandwidth reservation in cellular networks using road topology based mobility prediction," in Proceedings of IEEE INFOCOM, 2004.
6. Roland Zander and Johan M Karlsson, "Predictive and Adaptive Resource Reservation (PARR) for Cellular Networks," International Journal of Wireless Information Networks, vol. 11, no. 3, pp. 161-171, 2004.
7. R. Sutton and A. G. Barto, Reinforcement Learning. Cambridge, Massachusetts: The MIT press, 1998.
8. C. Yener, A. Rose, "Genetic algorithms applied to cellular call admission: local policies," IEEE Transaction on Vehicular Technology, vol. 46, no. 1, pp. 72–79, 1997.
9. M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, 1994.
10. T. K. Das, A. Gosavi, S. Mahadevan, and N. Marchalleck, "Solving semi-markov decision problems using average reward reinforcement learning," Management Science, vol. 45, no. 4, pp. 560–574, 1999.