PAPER

A Reinforcement Learning Approach for Admission Control in Mobile Multimedia Networks with Predictive Information[*]

José Manuel GIMÉNEZ-GUZMAN[†a)], *Nonmember*, **Jorge MARTÍNEZ-BAUSET**[†], *and* **Vicent PLA**[†], *Members*

SUMMARY We study the problem of optimizing admission control policies in mobile multimedia cellular networks when predictive information regarding movement is available and we evaluate the gains that can be achieved by making such predictive information available to the admission controller. We consider a general class of prediction agents which forecast the number of future handovers and we evaluate the impact on performance of aspects like: whether the prediction refers to incoming and/or outgoing handovers, inaccurate predictions, the anticipation of the prediction and the way that predictions referred to different service classes are aggregated. For the optimization process we propose a novel Reinforcement Learning approach based on the concept of afterstates. The proposed approach, when compared with conventional Reinforcement Learning, yields better solutions and with higher precision. Besides it tackles more efficiently the curse of dimensionality inherent to multimedia scenarios.

Numerical results show that the performance gains measured are higher when more specific information is provided about the handover time instants, i.e. when the anticipation time is deterministic instead of stochastic. It is also shown that the utilization of the network is maintained at very high values, even when the highest improvements are observed. We also compare an optimal policy obtained deploying our approach with a previously proposed heuristic prediction scheme, showing that plenty of room for technological innovation exists.

key words: *cellular mobile multimedia networks, admission control, optimization, predictive information*

1. Introduction

Session Admission Control (SAC) is a key traffic management mechanism in mobile multimedia 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 blocking 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, 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. In afterstates RL, decisions are taken based on the resulting state after the action is performed rather than on the current state at which the decision is taken. We show that, compared to conventional RL, afterstates RL achieves better solutions and does it with higher precision. Additionally, the afterstates RL is better sui-

[†] The authors are with the Dept. of Communications, Universidad Politécnica de Valencia (UPV), 46022 Valencia, Spain.

a) E-mail: jogiguz@upvnet.upv.es

[*] This work has been supported by the Spanish Ministry of Education and Science and European Comission (30% PGE, 70% FEDER) under projects TSI2005-07520-C03-03 and TEC2004-06437-C05-01.

ted to multiservice scenarios as it gives rise to a lower cardinality state space than conventional RL. In [8] the performance of an exact optimization approach based on dynamic programming was compared with the performance of the conventional and afterstates reinforcement learning approaches in a single service scenario. The conclusion was that the performance of SAC policies obtained by the afterstates approach was as good as those obtained by the exact approach.

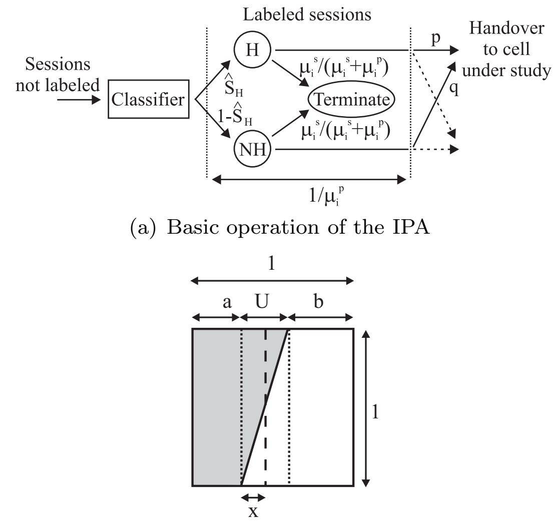
We consider a general multiservice scenario with available movement predictive information that feeds the SAC, and carry out a numerical evaluation to asses the impact of using afterstates RL on the one hand, and several aspects regarding the nature of the predictive information on the other. In an earlier version of the prediction scheme we were providing the optimization process only with state information of the neighbouring 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 [9] reached the same conclusion but using a genetic algorithm to find near-optimal policies. The predictive movement information considered is characterized by: its degree of certainty, whether it refers to incoming and/or outgoing handovers, the time frame at which it is forecasted to become effective, and the way that predictions related to different services are aggregated.

The contributions of this paper are three-fold: considering the use of movement predictive information in the SAC from an optimization perspective, evaluating the effect of the characteristics of that predictive information on the SAC performance and exploring the application of an afterstates RL technique to the SAC problem. Besides, we compare an optimal policy obtained deploying our approach with a previously proposed heuristic prediction scheme. The big performance difference between both approaches shows that innovative solutions for the design of SAC systems that make use of predictive movement information are still possible.

In Section 2 we describe the models of the system and of the different types of predictive information considered. The optimization framework and a description of the developed afterstates RL approach is presented in Section 3. A numerical study is provided in Section 4, which compares the RL approach based on afterstates with the conventional RL one and quantifies the impact on system performance of the different types of predictions explored. In Section 4 we also compare our proposed optimization approach to an heuristic one for the design of SAC policies. Finally, a summary of the paper and some concluding remarks are given in Section 5.

2. Model Description and Prediction System

We consider a single cell system and its neighbourhood,



(a) Basic operation of the IPA

(b) Basic parameters of the classifier

Fig. 1 IPA and classifier models.

where the cell has a total of C resource units and the neighbourhood C_p resource units, being the physical meaning of a unit of resources dependent on the specific technological implementation of the radio interface. A total of N different services are offered by the system. For each service new and handover session arrivals are distinguished so that there are N services and $2N$ arrival types.

For the sake of mathematical tractability we make the common assumptions of Poisson arrival processes and exponentially distributed random variables for cell residence time and session duration. The arrival rate for new (handover) sessions of service i is λ_i^{nc} (λ_i^{hc}) and a request of service i consumes b_i resource units, $b_i \in \mathbb{N}$. For a packet based air interface, b_i represents the effective bandwidth of the session [10], [11]. For service i, the session duration and cell residence rates are μ_i^s and μ_i^r respectively. The resource holding time in a cell for service i is also exponentially distributed with rate $\mu_i = \mu_i^s + \mu_i^r$ and the mean number of handovers per session is $N_i^h = \mu_i^r/\mu_i^s$. Without loss of generality, we will assume that only one session is active per mobile terminal (MT).

Given that the focus of our study was not the design of the prediction agent (PA), we used a model of it instead.

2.1 Prediction Agent for Incoming Handovers

An active MT entering the cell neighbourhood 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 ($\hat{S}^{\rm e}_H$) and the shaded area on the right of the threshold ($\hat{S}^{\rm e}_{NH}$). 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 $(\mu_i^p)^{-1}$, 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 $q \neq 1-p$. It can be shown that

$$1-p=\frac{\hat{S}^{\rm e}_H}{\hat{S}_H}=\frac{x^2}{(U(2S_H-U+2x))} \qquad (1)$$

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

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, i.e. they will abandon the cell before their ongoing session finishes, 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_i^r. The fraction of sessions that effectively execute an outgoing handover is given by $S_H^{\rm out}=\mu_i^r/(\mu_i^s+\mu_i^r)$. 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 (1) and (2), changing S_H by $S_H^{\rm out}$.

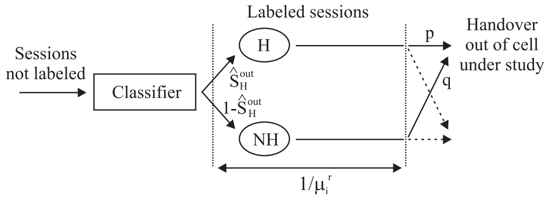


Fig. 2 Basic operation of the OPA.

2.3 Stochastic and Deterministic Time Prediction

In the prediction agents described in Sections 2.1 and 2.2, the amount of time elapsed since an active mobile terminal (MT) is deemed as "probably producing a handover" until the handover actually occurs is not predicted by the PA and we model it by an exponential random variable. This type of prediction is called stochastic prediction.

However, we can include into the PA a more precise knowledge of the future handover time instants for evaluating its impact 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. This prediction is called deterministic prediction. More precisely, in deterministic prediction both the IPA and OPA operate as before but they label the sessions T time units before the destination of the MT is definitive, i.e. now the IPA (OPA) informs of the number of incoming (outgoing) sessions that are finishing or producing a handover in less than T time units. 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.

3. Optimization by Reinforcement Learning

The information provided by the IPA and/or the OPA and the state of the cell (number of occupied resources) are used to find the optimal admission policy and its performance. We formulate the optimization problem as an infinite-horizon finite-state *Semi-Markov Decision Process* (SMDP) under the average cost criterion. SMDPs are a special kind of *Markov decision processes* (MDPs) appropriate for modeling continuous-time systems in which the time between decision epochs is not constant. Formally, a MDP can be defined as a tuple $\{\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{C}\}$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{P}$ is a state transition function and $\mathcal{C}$ is a cost function. The agent can control the state of the system by choosing actions a from $\mathcal{A}$, influencing in this way the state transitions. The transition function $\mathcal{P}:\mathcal{S}\times\mathcal{A}\rightarrow\mathcal{S}$ specifies the effect of taking an action at a given state.

The agent knows the state of the system s at any time and it chooses actions based only on the current state. We consider deterministic stationary Markovian policies, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, which defines the next action of the agent based only on the current state $\boldsymbol{x}$, i.e. an agent adopting this policy performs action $\pi(s)$ in state s. For the problems we consider, optimal stationary Markovian policies always exist.

We assume a bounded, integer-valued cost function $\mathcal{C} : \mathcal{S} \rightarrow \mathbb{N}$, and denote by $c(\boldsymbol{x}, a)$ the finite cost for executing action a in state $\boldsymbol{x}$. We define the total cost accumulated in the interval $[0, t]$ as

$$w^{\pi}(\boldsymbol{x}_0, t) = \sum_{m=0}^{t} c\big(x_m, \pi(x_m)\big)$$

If the environment is stochastic then $w^{\pi}(\boldsymbol{x}_0, t)$ is a random variable. Under the average cost criterion we seek to minimize the average expected cost rate over time t, as $t \rightarrow \infty$. When the system starts at state $\boldsymbol{x}$ and follows policy π, the average expected cost rate is denoted by $\gamma^{\pi}(\boldsymbol{x})$ and is defined as

$$\gamma^{\pi}(\boldsymbol{x}) = \lim_{t\rightarrow\infty} \frac{1}{t} E\left[w^{\pi}(\boldsymbol{x}, t)\right]$$

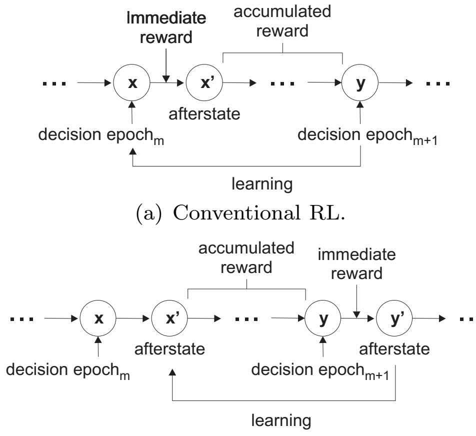
In a system like ours, it is not difficult to see that for every deterministic stationary policy the embedded Markov chain has a unichain transition probability matrix, and therefore the average expected cost rate does not vary with the initial state [12]. We call it the "cost rate" of the policy π, denote it by γ^{π} and consider the problem of finding the policy π^* that minimizes γ^{π}, which we name the optimal policy.

It can be shown that for our systems the cost structure is chosen so that the average expected cost rate represents a weighted sum of the loss rates

$$\gamma^{\pi} = \sum_{i=1}^{N} (\omega_i^n P_i^n \lambda_i^n + \omega_i^h P_i^h \lambda_i^h)$$

where ω_i^n (ω_i^h) is the relative weight associated to the blocking of a new (handover) request and P_i^n (P_i^h) is the blocking probability of new (handover) requests, both of service i. In general, $\omega_i^n < \omega_i^h$ since the loss of a handover request is less desirable than the loss of a new session setup request.

Decision epochs are those time instants in which we must select an action from the set of possible actions $\mathcal{A} := \{0 = \text{reject}, 1 = \text{admit}\}$. Given that no actions are taken at session departures, then only the arrival events are relevant to the optimization process. We select one of the $2N$ arrival types as the highest priority one, being its requests always admitted while free resources are available, and therefore no decisions are taken for them. The cost structure is defined as follows. At any decision epoch, the cost incurred by accepting any arrival type is zero and by rejecting a new (handover) request of service i is ω_i^n (ω_i^h). With this framework, further accrual of cost occurs when the system has to reject requests of the highest priority arrival type between two decision epochs.



(a) Conventional RL.

(b) Afterstates RL.

Fig. 3 RL approach.

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

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

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, γ^* is the average expected cost rate of the optimal policy, $w(\boldsymbol{x}, a)$ 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 $a = \pi(\boldsymbol{x})$.

3.1 Afterstates Reinforcement Learning

Intuitively, in systems as the one being considered, afterstates RL is based on the idea that what is relevant in the RL approach 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 RL and afterstates RL is shown in Fig. 3.

Being x_0 the number of resource units occupied in the cell under study and x_{in} (x_{out}) the number of resources occupied by sessions labeled as H by the IPA (OPA), the state spaces for the different scenarios are

$$\mathcal{S} := \{\boldsymbol{x} = (x_0, x_{in}) : x_0 \leq C; x_{in} \leq C_p\}$$

$$\mathcal{S} := \{\boldsymbol{x} = (x_0, x_{out}) : x_{out} \leq x_0 \leq C\}$$

$$\mathcal{S} := \{\boldsymbol{x} = (x_0, x_{in}, x_{out}) : x_{out} \leq x_0 \leq C; x_{in} \leq C_p\}$$

for the scenario that only considers the incoming handovers, for the scenario that only considers the outgoing handovers and when both incoming and outgoing prediction schemes are performed, respectively. In multimedia mobile cellular networks, the introduction of afterstates into the admission control optimization let us to reduce the cardinality of the state space, in comparison to the conventional RL approach [13]. For this last approach and for the incoming prediction, the state space was defined by $\mathcal{S} := \{\boldsymbol{x} = (x_0, x_{in}, k) : x_0 \leq C; x_{in} \leq C_p; 1 \leq k \leq (2N-1)\}$. As it can be seen, there exists an additional coordinate (k) related to the type of arrival, that can take $2N-1$ different values. In the RL afterstates approach, as we learn over the state reached immediately after the action is taken, the arrival type is not needed in the learning process, as it is already included in the afterstate. Therefore, the afterstates approach is independent on the number of services involved. This characteristic is specially important in systems with a high number of services, where RL based on afterstates tackles more efficiently the curse of dimensionality. Besides, as any RL optimization method, offers the important advantage of being a model-free method, i.e. transition probabilities and average costs are not needed in advance.

We deploy a modified version of the SMART algorithm [14] which follows an afterstates RL approach using a temporal difference method (TD(0)). The pseudo code of the proposed algorithm is shown in Fig. 4.

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 $(\omega_{reject} + h(\boldsymbol{y'}_{reject})) < h(\boldsymbol{y'}_{accept})$ 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. 3(b)).
    if $a$ is greedy:
        $W_T \leftarrow W_T + w_c(\boldsymbol{x'}, \boldsymbol{y}) + w(\boldsymbol{y}, a)$
        $T_T \leftarrow T_T + \tau$
        $\gamma \leftarrow W_T/T_T$
    $\boldsymbol{x'} \leftarrow \boldsymbol{y'}$
```

Fig. 4 SMART algorithm with afterstates.

4. Numerical Study

We assume a circular-shaped cell of radio r and a holed-disk-shaped neighbourhood with inner (outer) radio $1.0r$ ($1.5r$). The ratio of arrival rates of new sessions to the cell neighbourhood (ng) and to the cell (nc) is made equal to the ratio of their surfaces, $\lambda_i^{ng} = 1.25\lambda_i^{nc}$. The ratio of handover arrival rates to the cell neighbourhood from the outside of the system (ho) and from the cell (hc) is made equal to the ratio of their perimeters, $\lambda_i^{ho} = 1.5\lambda_i^{hc}$. Since the system is chosen to be in statistical equilibrium, the rate at which handover sessions enters the cell is equal to the rate at which handover sessions exits the cell (being both λ_i^{hc}), having

$$\lambda_i^{hc} = \mu_i^r/(\mu_i^r + \mu_i^s)\Big[(1 - P_i^n)\lambda_i^{nc} + (1 - P_i^h)\lambda_i^{hc}\Big]$$

Substituting P_i^h by $P_i^h = (\mu_i^s/\mu_i^r) \cdot [P_i^{ft}/(1 - P_i^{ft})]$, where P_i^{ft} is the probability of forced termination of a successfully initiated session, and after some algebra we get

$$\lambda_i^{hc} = (1 - P_i^n)(1 - P_i^{ft})N_i^h\lambda_i^{nc} \qquad (3)$$

Note that in our numerical experiments the values of the arrival rates are chosen to achieve realistic operating values for $P_i^n (\approx 10^{-2})$ and $P_i^{ft} (\approx 10^{-3})$. For such values, we approximate (3) as $\lambda_i^{hc} \approx 0.989N_i^h\lambda_i^{nc}$.

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 $p_m = p_0/(1 + u_m)$, where $u_m = m^2/(\varphi + m)$. A fixed value has been used for $p_0 = 0.2$ and varying values for φ depending on the size of the state space. The exploration of the state space is a common RL technique used to avoid being trapped at local minima. Due to the simulation-based nature of RL, each point in the figures represents the average of 10 different simulation runs initialized with different seeds.

Except in Section 4.3 and Section 4.4, where a higher number of services or only one is considered respectively, the scenario used in the numerical experiments is as follows. The number of services is $N = 2$, having the second service the highest priority. The rest of the parameters are: $C = 10$ and $C_p = 60$ resource units, $N_i^h = \mu_i^r/\mu_i^s = 1$, $\mu_i^r/\mu_i^p = 0.5$, $S_H = 0.4$, $x = U/2$ and $b_1 = 1$ and $b_2 = 2$ resource units. We also set $\mu_1 = \mu_1^s + \mu_1^r = 1$ and $\mu_2 = \mu_2^s + \mu_2^r = 3$. The arrival rates of new sessions to the cell are $\lambda_1^{nc} = 0.8\lambda_T$, $\lambda_2^{nc} = 0.2\lambda_T$, where $\lambda_T = 2$. The relative weights associated to the blocking of a new (handover) request are $\omega_1^n = 1$, $\omega_2^n = 4$, $\omega_1^h = 20$ and $\omega_2^h = 80$. In this scenario we have used $\varphi = 10^{12}$ when studying both incoming and outgoing prediction, and $\varphi = 10^{11}$ in the rest of the cases.

4.1 Learning Techniques

In this section we evaluate the performance of the afterstates RL approach with regard to the conventional RL. The advantage of using afterstates RL is two-fold: on the one hand, in Fig. 5(a) we can see the higher performance of the policies obtained when using afterstates in identical scenarios for different values of the PA uncertainty (U), obtaining a gain around 10%. In addition to the higher performance, in Fig. 5(b) it can also be observed that the relative width of the 95% confidence intervals is smaller when we use afterstates. As observed, the solutions obtained when deploying afterstates RL are better (γ^π is higher) and more precise (the relative width of the confidence interval is smaller). Therefore, in the rest of the paper the afterstates approach is used.

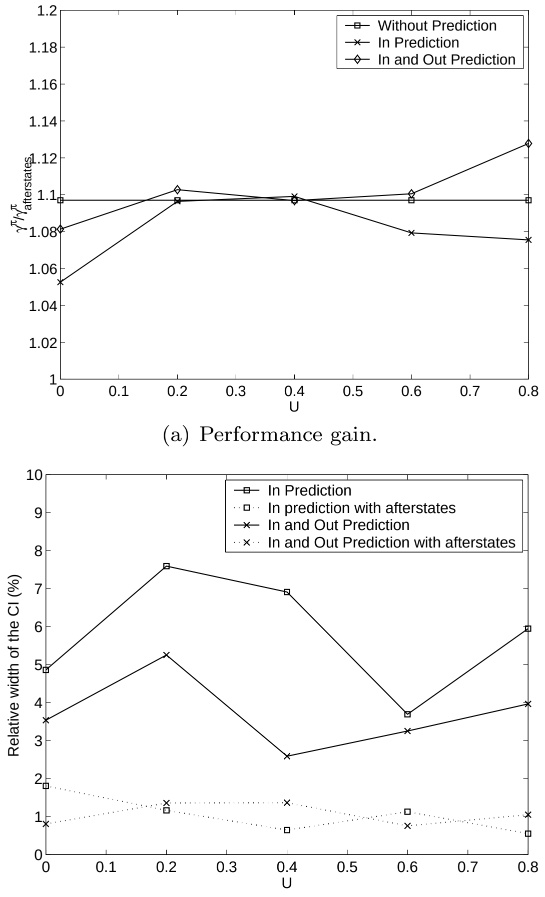
4.2 Type of Predictive Information

4.2.1 Stochastic Time Prediction

When introducing prediction, we evaluated the performance gain by the ratio $\gamma_{wp}^\pi/\gamma_p^\pi$, where γ_p^π (γ_{wp}^π) is the average expected cost rate of the optimal policy in a system with (without) prediction. Figure 6 shows the variation of the gain for different values of the uncertainty U when deploying 1) the IPA, 2) the OPA and 3) both the IPA and the OPA. As observed, using incoming handover prediction induces a gain and that gain decreases as the prediction uncertainty (U) increases. From Fig. 6 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. Without loss of generality, for a single service scenario 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

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



(a) Performance gain.

(b) Relative width of the confidence interval.

Fig. 5 Comparison of RL techniques in two service scenario.

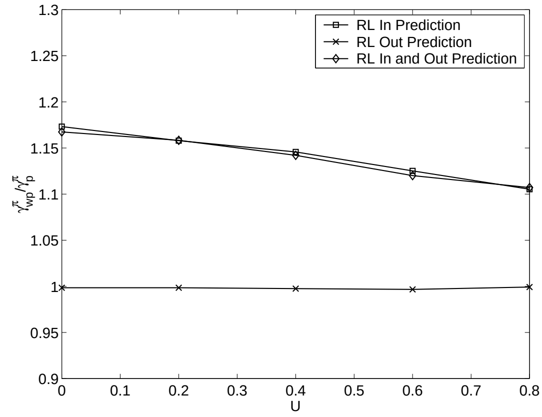


Fig. 6 Performance gain in stochastic handover prediction.

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

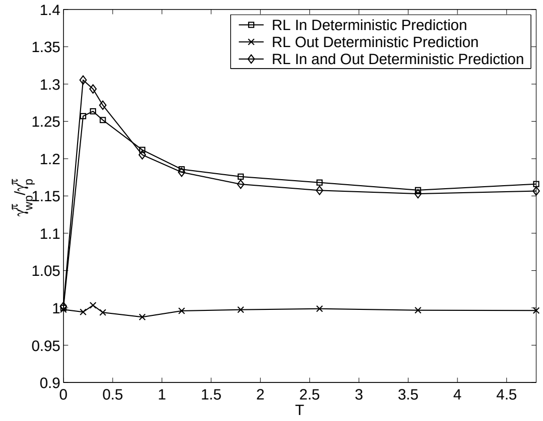


Fig. 7 Performance gain in deterministic handover prediction with U=0.2.

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.

4.2.2 Deterministic Time Prediction

Figure 7 shows the variation of the gain obtained using deterministic time prediction, for different values of T and $U = 0.2$. As observed, there exists an optimum value for T, which is close to the mean time between session arrivals, 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, because the labeling of sessions occur at the same time instants, i.e. when handover sessions enter the cell neighbourhood or the cell and when new sessions are initiated. When T is lower than its optimum value the gain also decreases, probably because the system has not enough time to react. When $T = 0$ the gain is null because there is no prediction at all. 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.

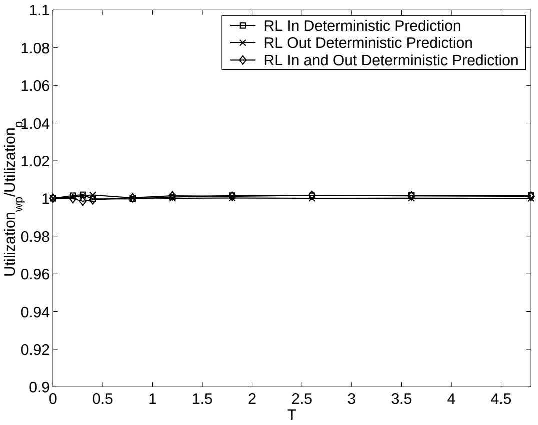


Fig. 8 Utilization gain when using deterministic prediction with U=0.

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 8 shows the ratio of the system resources utilization when not using prediction and when using prediction (utilization$_{wp}$/utilization$_{p}$). As the highest performance gain is obtained for $U = 0$, it could be expected to obtain also the highest utilization losses, but Fig. 8 shows that utilization is not reduced, what justifies the efficiency of our optimization approach.

4.2.3 Weighted prediction

Up to this point the information provided by the PA only anticipates the amount of resources required by the forecasted handovers but does not provide any information about their different priorities, which might be relevant to the decision process. This fact motivates us to study a scenario including weights dependent on the priority of the expected handovers. Taking the incoming prediction as a reference, now the state space would be defined by:

$$\mathcal{S} := \{\boldsymbol{x} = (x_0, x^w_{in}) : x_0 \leq C; x^w_{in} \leq \kappa C_p\}$$

where $\kappa = \omega^h_H/\omega^h_L$, being H (L) the highest (lowest) priority service and x^{ω}_{in} denotes the weighted number of forecasted handovers. In the scenario under study $\kappa = \omega^h_2/\omega^h_1 = 4$ and $x^{\omega}_{in} = x^{in}_1 + \kappa x^{in}_2$ Fig. 9 shows that substantially higher gains can be obtained at the expense of larger state spaces.

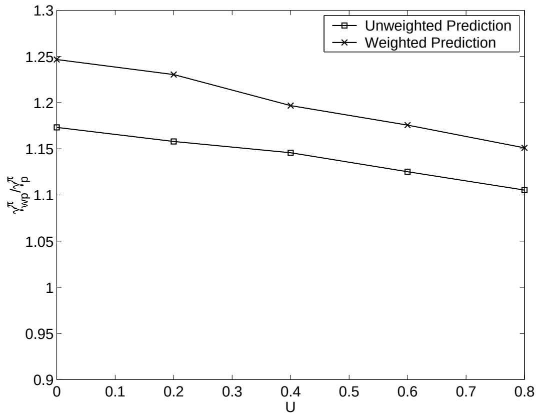


Fig. 9 Performance gain using weighted prediction and deploying input prediction.

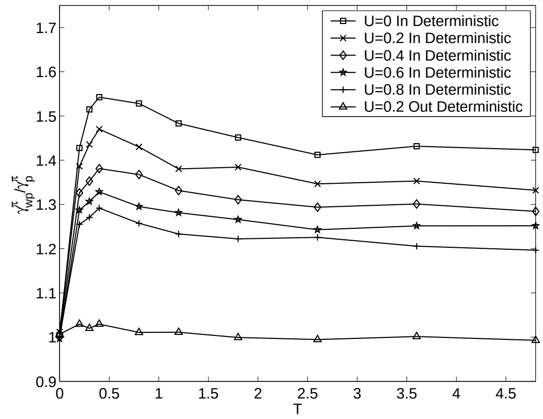


Fig. 10 Performance gain when using deterministic handover prediction in four services scenario.

4.3 Service Heterogeneity

Here we consider a scenario with a higher number of services and with a higher asymmetry among them. This setting in which high priority services demand more resources and represent a low part of the traffic aggregate, is appropriate for describing commercial multimedia wireless networks [10]. On the other hand, the experiments carried out in this section confirmed that the afterstates RL approach can handle scenarios with more services at no extra computational cost.

The parameters used for such a scenario have been: $N = 4$ (being $i = 4$ the highest priority service), $C = 20$ and $C_p = 120$ resource units, $N_i^h = \mu_i^r/\mu_i^s = 1$, $\mu_i^r/\mu_i^p = 0.5$, $\mu_i = \mu_i^s + \mu_i^r = 1$, $S_H = 0.4$, $x = U/2$ and $b_i = \{1, 2, 4, 6\}$ resource units per session. The arrival rates of new sessions to the cell are $\lambda_i^{nc} = \hat{f}_i \lambda_T$, where $\lambda_T = 2.8975$ and $\hat{f}_i$ can be obtained normalizing $f_i = \phi^{i-1}$, for $\phi = 0.2$. The relative weights associated to the blocking of a new (handover) request are $\omega_i^n = \{1, 4, 15, 60\}$ ($\omega_i^n = \{20, 80, 300, 1200\}$). In this scenario we have used $\varphi = 10^{12}$ in all the studied cases. Figure 10 shows the gain that is obtained in this scenario.

4.4 Comparative Evaluation

The performance of the SAC policy obtained by the RL optimization approach is compared to the performance of one of the predictive SAC schemes proposed in [15]. Although the predictive scheme proposed in [15] is applied in a scenario with two services, it is not a proper multiservice environment, as it does not distinguish between services neither in the admission process nor in the performance parameters. For that reason, the comparison with our approach (which can be applied to both single and multiservice scenarios) is done in a single service scenario.

Among the schemes proposed in [15] we choose the scheme AC1 instead of AC2 or AC3 because the same evaluation scenario deployed so far can be used to evaluate AC1, making the comparison more fair. Besides, the performance of AC1, AC2 and AC3 are quite similar for real operating conditions. Moreover, the performance of the three schemes was evaluated in [5] and the authors concluded that AC1 performs better.

The schemes proposed in [15] estimate the number of resource units that must be reserved in each cell for handovers occurring within a future time window of T_{est}. Let us denote by $C0$ the cell under study and by Ci its neighbouring cells. The number of resource units to be reserved in $C0$ for handovers arriving from neighbouring cell Ci is $B_{r,0}^i = \sum_{j \in Ci} b(C_{i,j}) p_h(C_{i,j} \rightarrow 0)$, where $b(C_{i,j})$ is the number of resource units required by the jth session in the ith cell and $p_h(C_{i,j} \rightarrow 0)$ is the estimated probability that session $C_{i,j}$ is handed over to cell 0 within T_{est} time units. The total amount of resource units that would be required to reserve in $C0$ are $B_{r,0} = \sum_{i \in A_0} B_{r,0}^i$, where A_0 is the set of indices of cell 0's neighbouring cells. Note that $B_{r,0}$ is a target, not the actual bandwidth reserved, since a cell may not have enough free resource units at the time the reservation is required.

In the evaluation scenario of our analysis we have the cell under study ($C0$) and its neighbourhood (Cg), therefore we estimate $B_{r,0}$ as $B_{r,0} = \sum_{j \in Cg} b(C_{g,j}) p_h(C_{g,j} \rightarrow 0)$, where $p_h(C_{g,j} \rightarrow 0)$ is the estimated probability that session $C_{g,j}$ is handed over to cell 0 within T_{est} time units.

The authors of [15] propose to estimate $p_h(C_{i,j} \rightarrow 0)$ by maintaining a data base that records historical movement patterns of users. Besides, each base station keeps track of each active mobile in its cell via the mobiles' *extant sojourn time*. The extant sojourn time of connection $C_{i,j}$, $T_{ext-soj}(C_{i,j})$, is the time elapsed since the active mobile with connection $C_{i,j}$ entered the cell Ci.

In the scenario scenario of our analysis, the probability $p_h(C_{g,j} \rightarrow 0)$ can be determined by:

$$p_h(C_{g,j} \rightarrow 0) = P(C_{g,j} \rightarrow 0) \cdot \\ P(\hat{t}_r \leq T_{est} + T_{ext-soj} | \hat{t}_r > T_{ext-soj})$$

where $\hat{t}_r$ is the residence time in the neighbourhood for those sessions that will execute a handover. The factor $P(C_{g,j} \rightarrow 0)$ represents the probability that those sessions in the neighbourhood that will execute a handover end up in $C0$, which is equal to the parameter S_H of the IPA model described in Section 2.1. The other factor represents the probability that a session that will execute a handover and which residence time in the neighbourhood is already longer than $T_{ext-soj}$, will issue the handover request in less than T_{est} time units from the current instant. To determine this second factor we recall the result of Lemma 1, which justifies that the distribution of the residence time in the neighbourhood for those sessions that will execute a handover $\hat{t}_r$ is also exponentially distributed with rate $\mu_p + \mu_s$. Therefore, given the memoryless property of the exponential distribution it follows that

$$P(\hat{t}_r \leq T_{est} + T_{ext-soj} | \hat{t}_r > T_{ext-soj}) = \\ P(\hat{t}_r \leq T_{est}) = 1 - e^{-(\mu_s + \mu_p) T_{est}}$$

As the number of historical records (N_{quad}) in the data base proposed in [15] grows, the sample value for $p_h(C_{i,j} \rightarrow 0)$, which is used in [15], will tend to its corresponding population value, i.e. the true probability value. In our comparative evaluation we consider that $N_{quad} \rightarrow \infty$ and therefore the movement estimation done in [15] is computed by

$$p_h(C_{g,j} \rightarrow 0) = S_H \big(1 - e^{-(\mu_s + \mu_p) T_{est}}\big)$$

Thus, the performance of the AC1 scheme that we obtain should be considered as an upper-bound of that obtained by the implementation deployed in [15].

The values of T_{est} for each cell are dynamically adjusted based on the measured forced termination ratio among a number of handovers recently observed, so as to meet an objective for the forced termination probability. When a new session arrives to $C0$, the AC1 scheme proposes to perform the following simple test: if $\sum_{j \in C_0} b(C_{0,j}) + b_{new} \leq C - B_{r,0}$ then it is admitted, otherwise it is rejected, where b_{new} is the number of resource units required by the new session.

Figure 11 compares the performance of the AC1 scheme with the performance of different policies obtained by the RL approach in a single service scenario with the following parameters: $C = 10$ and $C_p = 60$ resource units, $N_i^h = \mu_1^r / \mu_1^s = 1$, $\mu_1^r / \mu_1^p = 0.5$, $\lambda_1^n = 3.5$, $\mu_1 = \mu_1^s + \mu_1^r = 1$, $S_H = 0.4$ and $x = U/2$. For the optimization procedure using RL we have used a relative weight associated to the blocking of a new request of value $\omega_1^n = 1$ and a value of $\varphi = 10^{11}$ for the exploration phase. The value of ω_1^h is conveniently changed in order to obtain different values in the curves of Fig. 11. We also include the performance of a static policy that in this case is a single fractional guard channel policy. Although it might be surprising that a static policy performs better than the AC1 scheme, it should be pointed out that for the determination of the optimum number of guard channels it is required to know traffic parameters like arrival rates and mean channel holding time, which are not required by the AC1 scheme. It is clear that the performance of policies obtained by the RL approach outperform the AC1 scheme. In Fig. 12 it is also shown that the utilization of the system is also higher with the scheme that deploys predictive information and uses RL for computing the optimal admission control policy.

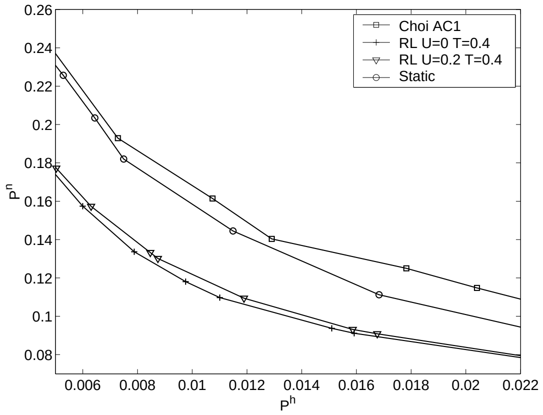


Fig. 11 Comparison of prediction schemes.

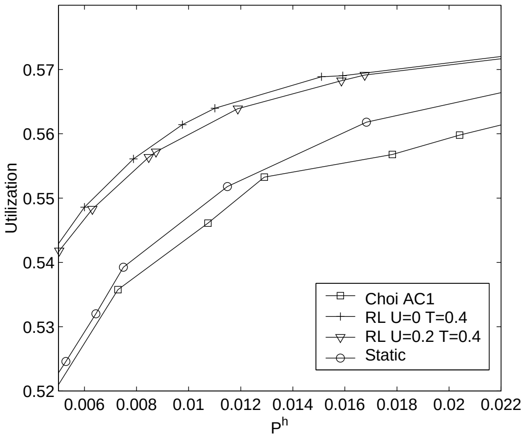


Fig. 12 Comparison of utilization with different prediction schemes.

5. Conclusions

In this paper we evaluate the performance gain that can be expected when the SAC optimization process is provided with predictive information related to incoming, outgoing and incoming and outgoing handovers together, in a multiservice 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 neighbourhood which 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 semi-Markov decision process, using Reinforcement Learning as solving methodology, having developed an afterstates based approach. We have shown that the developed approximation based on afterstates offers better solutions with higher precision than those results obtained without afterstates. Moreover, this approach is able to optimize admission control policies for multimedia scenarios regardless of the number of services being cursed in the network.

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. We also get higher performance gains when there is a distinction among the priority of the different handovers that are forecasted. Finally, we have compared our prediction scheme with the one proposed in [15], and the results show that our proposed methodology clearly outperforms that scheme in terms of blocking probabilities and utilization.

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.

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," Proceedings of IEEE PIMRC, 2002.

[4] V. Pla and V. Casares-Giner, "Optimal admission control policies in multiservice cellular networks," Proceedings of the International Network Optimization Conference (INOC), pp.466–471, Oct. 2003.

[5] W.S. Soh and H.S. Kim, "Dynamic bandwidth reservation in cellular networks using road topology based mobility prediction," Proceedings of IEEE INFOCOM, 2004.

[6] R. Zander and J.M. Karlsson, "Predictive and adaptive resource reservation (PARR) for cellular networks," International Journal of Wireless Information Networks, vol.11, no.3, pp.161–171, July 2004.

[7] R. Sutton and A.G. Barto, "Reinforcement Learning," Cambdridge, Massachusetts: The MIT press, 1998.

[8] J.M. Giménez-Guzmán, J. Martínez-Bauset and V. Pla, "An Afterstates Reinforcement Learning Approach to Optimize Admission Control in Mobile Cellular Networks," Wireless Systems and Network Architectures in Next Generation Internet, Matteo Cesana and Luigi Fratta (eds.), Lecture Notes in Computer Science (LNCS), vol.3883, pp.115–129, 2006.

[9] A. Yener and C. Rose, "Genetic algorithms applied to cellular call admission: local policies," IEEE Transactions on Vehicular Technology, vol.46, no.1, pp.72–79, 1997.

[10] S. Biswas and B. Sengupta, "Call admissibility for multirate traffic in wireless ATM networks," Proceedings of IEEE INFOCOM, pp.649–657, 1997.

[11] Jamie S.Evans and David Everitt, "Effective Bandwidth-Based Admission Control for Multiservice CDMA Cellular Networks," IEEE Transactions On Vehicular Technology, vol. 48, no. 1, pp. 36–46, January 1999.

[12] M.L. Puterman, Markov Decision Processes : Discrete Stochastic Dynamic Programming, John Wiley & Sons, 1994.

[13] V. Pla, J.M. Giménez-Guzmán, J. Martínez, and V. Casares-Giner, "Optimal bandwidth reservation in multiservice mobile cellular networks with movement prediction," IEICE Transactions on Communications, vol.E88-B, no.10, pp.4138–4141, Oct. 2005.

[14] 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.

[15] S. Choi and K.G. Shin, "Adaptive bandwidth reservation and admission control in QoS-sensitive cellular networks," IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 9, pp. 882–897, Sept. 2002.

José Manuel Giménez-Guzmán

Jorge Martínez-Bauset

Vicent Pla