Admission Control in UMTS Networks based on Approximate Dynamic Programming

Antonio Pietrabissa

Computer and System Science Department (DIS), University of Rome "La Sapienza", via Eudossiana 18, 00184 Rome, Italy

This paper presents a Connection Admission Control (CAC) algorithm for Universal Mobile Telecommunications System (UMTS) networks based on an Approximate Dynamic Programming (ADP) approach. To deal with the non-stationary environment due to the time-varying statistical characteristics of the offered traffic, the admission policy has to be computed periodically based on on-line measurements. If standard algorithms are used, the optimal policy computation is excessively time-consuming to be performed on-line. Thus, an ADP approach for the computation of a sub-optimal admission policy is proposed. The ADP approach is based (i) on the reduction of the policy space, and (ii) on an approximated state-space aggregation. Theoretical results and numerical simulations show the effectiveness of the proposed approach, which is currently being implemented in a real UMTS testbed.

Keywords: Approximate Dynamic Programming (ADP), Connection Admission Control (CAC), Markov Decision Process (MDP), Universal Mobile Telecommunications System (UMTS).

1. Introduction

This document presents a Connection Admission Control (CAC) strategy for wide-code division multiple access (WCDMA) networks based on an Approximate Dynamic Programming (ADP) approach. CAC consists in refusing a new connection if the addition of its traffic would lead to an unacceptable degradation of that or previously accepted traffic. The admission control problem has been successfully described as a Markov Decision Process (MDP), based on the fact the decision to accept or reject a call impacts on whether future calls will be accepted or not (see [1]). As shown in Section 2, Dynamic Programming (DP) algorithms can be used to compute the optimal admission policy once the CAC problem is represented as a MDP.

Two problems arises when DP algorithms are proposed for the implementation in real networks: (i) the "curse of dimensionality", which is the exponential growth of the state dimension as the number of links increases [3]; (ii) the stationary hypothesis underlying the MDP, which is not realistic due to non-stationary traffic characteristics.

With regard to the curse of dimensionality, the WCDMA scenario is not incompatible for a DP approach since, from the CAC viewpoint, each cell (a cell identifies a group of terminals transmitting to/receiving from the same base station) is almost independent of each other—the inter-cell interference is usually modelled by a pre-defined constant (see Section 3). Thus, the WCDMA CAC problem is essentially a single-link problem, and the dimension of the

---

Correspondence to: A. Pietrabissa, E-mail: pietrabissa@dis.uniroma1.it

1This work was supported by the EuQoS project ([6]), funded under the European Commission IST (Information Society Technologies) 6th Framework Programme, coordinated by

Received 12 August 2006; Accepted 9 November 2007

Recommended by J. Lunze, A.J. Van der Schaft
state space remains tractable [22,5]. Moreover, the single-link case is meaningful also for multi-link networks: in fact, the multi-link case can be reduced to the single-link case by assuming the link independency approximation [21,23,18].

As regards the latter problem, i.e., the non-stationary environment, it requires the admission policy to be update periodically, based on on-line measures of traffic statistics. Thus, even if we are considering a single-link case, the policy computation cost becomes critical. Therefore, a novel ADP algorithm specifically tailored for the admission control problem in communication networks is proposed.

ADP approaches are used to fast compute sub-optimal policies by introducing some approximations in the model—for instance by performing state aggregation or by reducing the policy space (see [2] for a survey on ADP and comprehensive references).

1.1. Related Work

A significant number of MDP-based admission control algorithms were proposed in the literature, both for terrestrial and wireless networks.

In all terrestrial networks, the above-mentioned link independency hypothesis is assumed. For instance, in [21] the CAC problem in optical networks is dealt with: the problem is formulated as a single link MDP and solved via the value iteration algorithm [7]. In [23], the policy iteration algorithm [7] is used to compute the optimal admission policy in ATM (Asynchronous Transfer Mode) terrestrial networks. Other examples can be found in [28,13,10]. Also MDP-based CAC for CDMA networks have been analyzed in the literature: for example, in [22], the multi-service admission control problem with fairness guarantees is solved by formulating the MDP problem as a Linear Programming (LP) one (as described in [7,27]); in [31], the DP approach is aimed at maximizing the revenue; in [34], the average data throughput is maximized under a blocking probability constraint; in [4], a fairness constraint is introduced in the LP, which enforces the difference between the blocking probabilities of different classes to be lower than a certain value; other examples can be found in [28,13,20].

In [15] the single link problem is modelled as a MDP, and a state grouping technique is developed; the technique is based on the concept of bandwidth quantization [18] and is used to evaluate “product-form” policies [14]; even if well-known policies, such as the greedy (or complete-sharing) one, have product-form distributions, this is not generally true for all the policies (see [18] for some examples). The novelties proposed in this paper are (i) that the proposed state space reduction technique is based on an ADP approach and is not limited to product-form policies, and (ii) that the state aggregation is used within a policy computation (and not only evaluation) algorithm tailored to an on-line implementation.

1.2. Paper Outline

The paper is organized as follows: Section 2 presents the proposed CAC approach; Section 3 describes the Universal Mobile Telecommunications System (UMTS) scenario and the CAC role; Section 4 shows numerical simulation results; finally, in Section 5 the conclusions are drawn and ongoing and future work is outlined.

2. MDP Connection Admission Control

Let us consider a single link in a generic network supporting \( C \) classes of service, each one requiring a load share \( \Delta L^{(c)}, c = 1, \ldots, C \). The maximum link load is denoted with \( \eta_{\text{MAX}} \). The network can be represented by a discrete-time system, whose state is the number of on-going connections of each class. Under the assumption that each on-going connection is compliant with its declared parameters, the controller has a perfect knowledge of the state, since, at time \( t \), the load \( \eta(t) \) is given by \( \sum_{c=1}^{C} \Delta L^{(c)} n^{(c)}(t) \), where \( n^{(c)}(t) \) is the number of connections of service class \( c \), on-going at time \( t \).

Let us define the state \( x(t) \) at time \( t \) as follows:

\[
x(t) = \left(n^{(1)}(t), n^{(2)}(t), \ldots, n^{(C)}(t)\right) \tag{1}
\]

The system is sampled with sample time \( \gamma \), defined in Section 2.1, and has the following dynamics:

\[
x(t+1) = f(x(t), u(t), z(t)), \tag{2}
\]

where \( u(t) \) is the control action of the CAC controller and the disturbance \( z(t) \) represents the connection attempt and terminations, characterized as follows:

(i) for each class \( c \), connection attempts are distributed according to a Poisson process with mean arrival frequency \( \lambda^{(c)} \);
(ii) the connection holding time of class \( c \) is exponentially distributed with mean termination frequency \( \mu^{(c)} \).

\footnote{Under the greedy policy, a connection is accepted unless the maximum link bandwidth is exceeded.}
Even if Poisson connection attempts and exponential connection holding time are widely used in the literature, and even if they are adequate for modeling voice users, they seem not to be fully justified for the new traffic services: in this respect, further research is needed in the area of Markov regenerative decision processes (see [26,12]).

The control action \( u(t) \) is the admission decision of the controller, and is relevant only at connection attempts. The control action is computed based on the current state \( x(t) \):

\[
u(t) = g(x(t)). \tag{3}\]

Figure 1 shows the system model. Given the model of the figure, the objective is to find the optimal controller \( u^*(t) = g^*(x(t)) \), which maximizes an appropriate reward function.

In Section 2.1, the network is modelled as a MDP based on the discrete model of Fig. 1; in Section 2.2, the procedure to fast compute a sub-optimal policy is proposed.

2.1. MDP model of the network

To model the discrete-time system of Fig. 1 as a MDP, the state space \( S \), the action space \( A \), the transition probability matrix \( T \) and the one-step reward function \( R \) have to be defined.

**State Space S**

The feasible states are the ones for which the following equation holds:

\[
\sum_{c=1}^{C} n_c(t) \Delta L_c(t) \leq \eta_{\text{MAX}}. \tag{4}\]

Let us assume that the finite number of feasible states identified by Eq. (4) is \((M + 1)\); hereafter, the feasible states will be denoted as follows:

\[
x_i = (n_i^{(1)}, n_i^{(2)}, \ldots, n_i^{(C)}), \quad i = 0, \ldots, M, \tag{5}\]

where \( n_i^{(c)} \) is the number of on-going connections of class \( c \) in state \( x_i \). The state space \( S \) consists of the set of the \((M + 1)\) feasible states:

\[
S = \left\{ x_i = (n_i^{(1)}, n_i^{(2)}, \ldots, n_i^{(C)}) \mid \sum_{c=1}^{C} n_i^{(c)} \Delta L_c(t) \leq \eta_{\text{MAX}} \right\} \tag{6}\]

\[
= \{ x_i, \quad i = 0, \ldots, M \}. \]

The assumption required to model the system as a MDP is the Markovian property, stating that the current state retains all relevant information independently of the past history of the system.

**Action Space A**

In the generic state \( x_i = (n_i^{(1)}, \ldots, n_i^{(C)}) \epsilon S \), if a connection attempt of class \( c \) occurs, the controller might decide to block the connection—and the state remains the same—or to accept it, provided that \( x_i + \delta_c \epsilon S \). By defining \( \delta_c \) as a vector of zeros with the \( c^{th} \) component equal to 1, it follows that \( x_i = x_i + \delta_c \). Let us denote such decision as \( u_i^{(c)} \), and let us associate the value 1 if the decision is to accept the new call, 0 if it is to reject it; no decision can be taken on connection terminations. The action space \( A \) is then defined as follows:

\[
A = \left\{ u_i = (u_i^{(1)}, \ldots, u_i^{(C)}) \mid u_i^{(c)} = \begin{cases} 1 & \text{if } x_i + \delta_c \epsilon S; c = 1, \ldots, C; \ i = 0, \ldots, M \end{cases} \right\}. \tag{7}\]

The maximum number of possible actions in a given state is \( 2^C \); hereafter, the \( k^{th} \) action will be denoted with \( a_k \). For instance, if \( C = 2 \) the following four actions can be taken: \( a_1 = [0, 0] \), reject all; \( a_2 = [0, 1] \), admit class 2 only; \( a_3 = [1, 0] \), admit class 1 only; \( a_4 = [1, 1] \), admit all. Not all the actions are feasible in every state: the actions \( a_k \) involving the admission of a class \( c \) connection are not feasible in the states \( x_i \epsilon S \) such that \( x_i + \delta_c \not\epsilon S \). Let us define the set \( A_i \) as the set of indexes of the feasible actions \( a_k \) in state \( x_i \), \( i = 0, \ldots, M \). For each state \( x_i \), \( A_i \) has at least one element, since the “reject all” action is always feasible. Hereafter, we will refer to the action \( a_k \epsilon A_i \) as the greedy action in state \( x_i \).

The controller \( g(x) = u = (u_1, \ldots, u_M) \) determines the admission policy, i.e., the decisions undertaken by the controller when a new connection arrives, based on the current state \( x \); \( g(x) = u \) maps the state \( x_i \epsilon S \) to the control action \( u_i \). The policy \( u \) is feasible if
The transition frequencies are straightforwardly inferred from the above-stated assumptions on \(z(t)\) and from the adopted policy \(u\):

- the frequency \(\varphi_{ij}\) of the transition between the states \(x_i \in S\) and \(x_j = x_i - \delta_c\) is equal to \(n_i^{(c)} \mu_i^{(c)}\) if \(x_j \in S\), to 0 otherwise;
- the frequency \(\varphi_{ij}\) of the transition between the states \(x_i \in S\) and \(x_j = x_i + \delta_c\) is equal to \(u_i^{(c)} \lambda_i^{(c)}\) if \(x_j \in S\), to 0 otherwise; note that this frequency depends on the policy \(u\);
- all other transitions between states are impossible, assuming that connection attempts and terminations are not simultaneous.

To apply DP algorithms, we have to obtain an uniform, discrete-time Markov chain. A standard method is to define the transition probabilities as follows [3]:

(i) Divide the transition frequencies of each state by the following constant (uniformization):

\[
\gamma = \max_{i=0, \ldots, M} \left\{ \sum_{c=1}^{C} \left( \lambda_i^{(c)} + n_i^{(c)} \mu_i^{(c)} \right) \right\},
\]

thus obtaining the following transition probabilities for the state \(x_i\):

- the probability \(p_{ij}\) of the transition between the states \(x_i \in S\) and \(x_j = x_i - \delta_c\) is given by:

\[
p_{ij} = \begin{cases} 
\frac{n_i^{(c)} \mu_i^{(c)}}{\gamma} & \text{if } x_j \in S; \ i = 0, \ldots, M; \ c = 1, \ldots, C, \\
0 & \text{otherwise}
\end{cases}
\]

which depends on the number of on-going connections of class \(c\) in state \(x_i\);
- the probability \(p_{ij}\) of the transition between the states \(x_i \in S\) and \(x_j = x_i + \delta_c\) is given by:

\[
p_{ij} = \begin{cases} 
\frac{\lambda_i^{(c)} \mu_i^{(c)}}{\gamma} & \text{if } x_j \in S; \ i = 0, \ldots, M; \ c = 1, \ldots, C, \\
0 & \text{otherwise}
\end{cases}
\]

which depends on the decision \(u_i = g(x_i)\) associated to the state \(x_i\);

(ii) Add a transition from the state \(x_i\) to itself (fictitious event), with probability \(p_{ii}\) equal to:

\[
p_{ii} = 1 - \sum_{j=1, \ldots, M, j \neq i} p_{ij}; \ i = 0, \ldots, M.
\]

**Reward Function \(R\)**

For CAC algorithms, the most obvious evaluation parameter is the link utilization. Since the generic state is specified by the number of on-going connections of each class \(c\), given the load share of each class \(\Delta L_i^{(c)}\) we can define the one-step reward associated to \(x_i\) as the load occupied by the on-going accepted connections; in this case, the one-step reward when the system evolves from \(x_i\) to \(x_j\) under policy \(u\) depends on \(x_j\) only:

\[
r_{ij} = \sum_{c=1}^{C} n_i^{(c)} \Delta L_i^{(c)}, \quad i, j = 0, \ldots, M.
\]

As specified below, the problem can be formulated as a LP problem (see [27]). In this case, the reward function can allow for more complex optimizations.

The described model \(\{S, A, T, R\}\) is a MDP; the generic element \(u_i^{(c)}\) of the policy \(u \in U\) is the control variable computed by the controller, and represents the decision to accept or not a connection of class \(c\) when the system is in state \(x_i \in S\).

Figure 2 shows a generic state with its transition probabilities in the case with \(C = 2\).

**Optimal policy \(u^*\) and greedy policy \(u_G\)**

Once the MDP is defined, the problem is to find the optimal policy \(u^*\) with respect to a certain cri-
In this paper, we consider the expected average reward\(^3\):

\[
R(u) = E_u\left\{ \sum_{t=1}^{\infty} r(x(t)) \right\},
\]

where \(x(t)\) is the state visited at time \(t\) and \(E_u\) denotes the expected value under policy \(u\).

The expected reward (14) can be computed as follows:

\[
R(u) = \sum_{i=0}^{M} r_i \pi_i(u),
\]

where \(\pi_i(u)\) is the stationary probability that the process is in state \(x_i\) under policy \(u\). Given a policy \(u\), the stationary probabilities can be obtained by solving a system of \((M+1)\) equations (see [7]) or by using approximated methods (see [30]); in particular, a closed form exists for the computation of the stationary probabilities of the greedy policy (see [15]).

Several DP algorithms exist which solve the defined problem (see [7,3]). In this paper, we are interested in the LP formulation [7,27]:

Maximize \( R = \sum_{i=0}^{M} \sum_{k \in A_i} r_i y_{ik} \)

subject to

(i) \( \sum_{k \in A_i} y_{ik} - \sum_{j=0}^{M} \sum_{k \in A_j} y_{jk} p_{ij}(a_k) = 0, \quad j = 0, \ldots, M-1; \)

(ii) \( \sum_{i=1}^{M} \sum_{k \in A_i} y_{ik} = 1; \)

and \( y_{ik} \geq 0, \quad i = 1, \ldots, M, \quad k \in A_i. \)

(16)

where \( A_i \) is the set of feasible actions in state \(x_i\), and \( y_{ik}\) is the generic unknown, representing the probability that the system is in state \(x_i\) and action \(a_k\) is chosen. It follows that \( \pi(x_i) = \sum_{k \in A_i} y_{ik} \) and, thus, that the LP reward function \( R \) is the expected average reward (14).

The solution of the LP generates the optimal policy \( u^* \) computed as follows:

\[
u_{i}^{(c)} = \sum_{k \in A_i} D_{ik} a_k,
\]

where \( D_{ik} \) is the probability of choosing action \(a_k\) when the system is in state \(x_i\):

\[
D_{ik} = \frac{y_{ik}}{\sum_{k \in A_i} y_{ik}}.
\]

\(\) Generally, the optimal solution of the LP originates randomized policies: the decision \( u_{i}^{(c)} \) is a real number between 0 and 1 and represents the probability of accepting the new call.

The greedy policy \( u_{i}^{(c)} \) is defined as follows: the decision \( u_{i}^{(c)} = (u_{i}^{(1)}, \ldots, u_{i}^{(C)}) \) associated to the state \(x_i, i = 0, \ldots, M\), is to accept the connection whenever it is possible:

\[
u_{i}^{(c)} = \begin{cases} 
1 & \text{if } x_i + \delta_c \in S, \quad c = 1, \ldots, C, \quad i = 0, \ldots, M. \\
0 & \text{otherwise}
\end{cases}
\]

(19)

2.2. Restricted MDP and Aggregated MDP

In this Section, the restricted MDP (Section 2.2.1) and the aggregated MDP (Section 2.2.2) are defined. Hereafter, the MDP defined in Section 2.1 will be referred to as the original MDP.

2.2.1. Restricted MDP

The restricted policy approach is extremely attractive for the admission control problem; in fact, when the link load is scarce, it is reasonable to accept all the new connections, independently of their class of service. Thus, if we restrict the policy space by forcing the admission of all the connections when the load is below a certain threshold, we are likely to include the optimal policy in the restricted policy space.

Let us number the states in decreasing order with respect to the occupied load (e.g., \(x_M\) is the empty state), and let us consider the following partition of the state space, including the first \(N\) states (the most loaded ones):

\[
\tilde{S} = \left\{ x_i = \left( n_i^{(1)}, n_i^{(2)}, \ldots, n_i^{(C)} \right), \right. \\
\left. \quad \times \left| \eta_{LOW} < \sum_{c=1}^{C} n_i^{(c)} \Delta L^{(c)} \leq \eta_{MAX} \right. \right\} \\
\left. \right\} = \{x_0, \ldots, x_{N-1}\} \subseteq S
\]

(20)

where \(1 \leq N \leq M\) and \(\eta_{LOW}\) is a load threshold such that \(0 \leq \eta_{LOW} \leq \eta_{MAX}\).

Let us consider the transition probability representing a connection request: the restricted policy approach is characterized by the fact that connections are always accepted in states \(x_N, \ldots, x_M\), i.e., in the states whose load is lower than or equal to the load
threshold $\eta_{LOW}$. The transition $p_{ij}$ between the states $x_i \in S$ and $x_j = x_i + \delta_i$ is then given by:

$$
p_{ij} = \begin{cases} 
\frac{\lambda(c)}{\gamma} u_i(c) & \text{if } x_i \in \bar{S} \text{ and } x_j \in S \\
\lambda(c) & \text{if } x_i \in (S - \bar{S}) \text{ and } x_j \in S \\
0 & \text{otherwise}
\end{cases}
$$

(21)

The transition probabilities $p_{ij}$ for $i = N, \ldots, M$ and $j = 0, \ldots, M - 1$, are independent of the policy. Since also the transition probabilities representing a connection termination—see Eq. (10)—are independent of the policy, the restricted action space is defined as follows:

$$\bar{A} = \left\{ u_i = (u_i^{(1)}, \ldots, u_i^{(C)}) | u_i^{(c)} = \{0, 1\} \text{ if } x_i + \delta_i \in \bar{S}, \\
u_i^{(c)} = 0 \text{ if } x_i + \delta_i \notin \bar{S}; c = 1, \ldots, C; i = 0, \ldots, N - 1 \right\}.
$$

(22)

Thus, the restricted policies are associated only to the states in $\bar{S} = \{x_0, \ldots, x_{N-1}\}$. The restricted policy space is defined as the set of all the feasible restricted policies:

$$\bar{U} = \{ \bar{u} = (u_0, \ldots, u_{N-1}) | u_i \in \bar{A} \}.
$$

(23)

The following Theorem 1 shows that the expected reward of the restricted MDP $\{\bar{S}, \bar{A}, \bar{T}, R\}$ is upper-bounded by the expected reward of the original MDP under the optimal policy and is at least equal to the expected reward of the original MDP under the greedy policy:

**Theorem 1:** By denoting with $u^* \in U$ and $u_G \in U$ the optimal policy and the greedy policy of the original MDP and with $\bar{u}^* \in \bar{U}$ the optimal policy of the restricted MDP, the following relation holds:

$$R(u_G) \leq R(\bar{u}^*) \leq R(u^*)
$$

(24)

**Proof:** Let us consider the restricted MDP under the generic restricted policy $\bar{u} = (\bar{u}_0, \ldots, \bar{u}_{N-1}) \in \bar{U}$. Considering Eq. (19), by comparing Eqs. (11) and (21), it is evident that the restricted MDP is identical to the original MDP under the policy $u' \in U$ such that the decisions associated to the non-aggregated states $u_{G,i}$, $i = N, \ldots, M$ are to accept the connection whenever it is possible:

$$u' = (\bar{u}_0, \ldots, \bar{u}_{N-1}, u_{G,N}, \ldots, u_{G,M}).
$$

(25)

Three consequences follow:

1. The restricted MDP under the restricted greedy policy $\bar{u}_G = (u_{G,0}, \ldots, u_{G,N-1}) \in \bar{U}$ is identical to the original MDP under the greedy policy $u_G = (u_{G,0}, \ldots, u_{G,M}) \in U$, and the following relation holds:

$$R(u^*) \geq R(\bar{u}_G) = R(u_G).
$$

(26)

2. If the optimal policy $u^* = (u_0^*, \ldots, u_M^*)$ of the original MDP is such that $u^* = (u_0^*, \ldots, u_{N-1}^*, u_{G,N}, \ldots, u_{G,M})$, it follows that $\bar{u}^* = (u_0^*, \ldots, u_{N-1}^*)$ is the optimal policy for the restricted MDP:

$$R(\bar{u}^*) = R(u^*) \text{ if } u^* = (u_0^*, \ldots, u_{N-1}^*, u_{G,N}, \ldots, u_{G,M}).
$$

(27)

In fact, by contradiction, if another restricted policy $\bar{u} = (u_0, \ldots, u_{N-1})$ were such that $R(\bar{u}) > R(\bar{u}^*)$, the policy $u' = (u_0^*, \ldots, u_{N-1}^*, u_{G,N}, \ldots, u_{G,M})$ would be such that $R(u') > R(u^*)$.

3. If the optimal policy $u^* = (u_0^*, \ldots, u_M^*)$ of the original MDP is such that $u^* \neq (u_0^*, \ldots, u_{N-1}^*, u_{G,N}, \ldots, u_{G,M})$, it follows that, given the optimal restricted policy $\bar{u}^* = (\bar{u}_0, \ldots, \bar{u}_{N-1})$, the policy $u' = (\bar{u}_0^*, \ldots, \bar{u}_{N-1}^*, u_{G,N}, \ldots, u_{G,M})$ is a sub-optimal policy for the original MDP:

$$R(\bar{u}^*) > R(u') < R(u^*)
$$

if $u^* \neq (u_0^*, \ldots, u_{N-1}^*, u_{G,N}, \ldots, u_{G,M})$.

(28)

From Eqs. (26), (27) and (28) Eq. (24) follows. 

\[\square\]

2.2.2. Aggregated MDP

In this Section, we will consider the aggregation of the restricted MDP defined in Section 2.2.1 under the restricted policy $\bar{u} \in \bar{U}$. The $(M - N)$ states $x_i \in (S - \bar{S})$ are aggregated into a single state $\bar{x}_N$. The aggregated state space $\bar{S}$ consists of the set of the states $x_0, \ldots, x_{N-1}$ plus $\bar{x}_N$:

$$\bar{S} = \left\{ x_i = (n_i^{(1)}, n_i^{(2)}, \ldots, n_i^{(C)}) \\
\times \eta_{low} < \sum_{c=1}^{C} n_i^{(c)} \Delta L^{(c)} \leq \eta_{hh} \right\} \cup \bar{x}_N
$$

(29)

The decision associated to the state $\bar{x}_N$ is to accept the connection requests whenever it is possible (greedy
policy); thus, the aggregated action space and the aggregated policy space are equal to the restricted ones:

\[
\tilde{A} = \left\{ \tilde{a}_i \left(1 \right), \ldots, \tilde{a}_i \left(M \right) \right\} \mid \tilde{a}_i = \left(0, 1 \right) \text{if } x_i + \delta_i \in \tilde{S}, \\
\tilde{u}_i = \left(0 \right) \text{if } x_i + \delta_i \notin \tilde{S}; c = 1, \ldots, C; i = 0, \ldots, N - 1 \right\} = \tilde{A}. 
\] (30)

\[
\tilde{U} = \left\{ u = (u_0, \ldots, u_{N-1}) \right\} \mid u_i \in \tilde{A} = \tilde{U}. 
\] (31)

Considering the stationary probabilities \(\pi_i(\tilde{u})\) \(i = 0, \ldots, M\) of the restricted MDP under policy \(\tilde{u}\), in the associated aggregated MDP the stationary probabilities \(\tilde{\pi}_i(\tilde{u})\) \(i = 0, \ldots, N\) are straightforwardly obtained as follows:

\[
\begin{align*}
\tilde{\pi}_i(\tilde{u}) &= \pi_i(\tilde{u}), & i &= 0, \ldots, N - 1 \\
\tilde{\pi}_N(\tilde{u}) &= \sum_{i=0}^{M} \pi_i(\tilde{u}). 
\end{align*} 
\] (32)

The transition probabilities to/from \(\tilde{x}_N\) are computed from Eqs. (10), (11) and (21):

\[
\begin{align*}
\tilde{p}_{ij}(\tilde{u}) &= p_{ij}(\tilde{u}), \\
\tilde{p}_{IN} &= \sum_{j=N}^{M} p_{ij}, & i &= 0, \ldots, N - 1 \\
\tilde{p}_{NJ}(\tilde{u}) &= \sum_{i=0}^{M} \pi_i(\tilde{u}) \pi_{Nj}(\tilde{u}), & j &= 0, \ldots, N - 1 \\
\tilde{p}_{NN}(\tilde{u}) &= 1 - \sum_{j=0}^{N-1} \tilde{p}_{NJ}. 
\end{align*} 
\] (33)

which define the transition matrix \(\tilde{T}(\tilde{u})\) (which depends on \(\tilde{u}\)).

Given the restricted policy \(\tilde{u}\) and Eq. (33), let us define the reward function of the aggregated MDP as follows:

\[
\tilde{R}(\tilde{u}) = \sum_{i=0}^{N} \tilde{r}_i(\tilde{u}), \text{ where } \tilde{r}_i(\tilde{u}) = \begin{cases} r_i & i = 0, \ldots, N - 1 \\ \sum_{j=N}^{M} r_{ij} \pi_j(\tilde{u}) & i = N. \end{cases} 
\] (34)

**Theorem 2:** The aggregated MDP under policy \(\tilde{u}\), \(\{\tilde{S}, \tilde{A}, \tilde{T}(\tilde{u}), \tilde{R}(\tilde{u})\}\), generates the same expected reward of the restricted MDP under policy \(\tilde{u}\):

\[
R(\tilde{u}) = \tilde{R}(\tilde{u}). 
\] (35)

**Proof:** By computing the stationary reward (15) of the aggregated MDP under policy \(\tilde{u}\), and by using Eqs. (32) and (34), the stationary reward of the restricted MDP under policy \(\tilde{u}\) is obtained:

\[
\tilde{R}(\tilde{u}) = \sum_{i=0}^{N} \tilde{r}_i(\tilde{u}) = \sum_{i=0}^{N-1} \tilde{r}_i(\tilde{u}) + \tilde{r}_N(\tilde{u}) \\
= \sum_{i=0}^{N-1} \pi_i(\tilde{u}) r_i + \sum_{i=0}^{M} \pi_i(\tilde{u}) r_i \\
= \sum_{i=0}^{M} \pi_i(\tilde{u}) r_i \\
= R(\tilde{u}), 
\] (36)

which proves the theorem. \(\square\)

**Corollary 1:**

1. If \(\tilde{u}^* \in \tilde{U}\) is the optimal policy for the restricted MDP, it is the optimal policy for the aggregated MDP \(\{\tilde{S}, \tilde{A}, \tilde{T}(\tilde{u}^*), \tilde{R}(\tilde{u}^*)\}\).

2. If \(u^* \in \tilde{U}\) is the optimal policy for the aggregated MDP \(\{\tilde{S}, \tilde{A}, \tilde{T}(\tilde{u}), \tilde{R}(\tilde{u})\}\), \(\tilde{u}\) is the optimal policy for the restricted MDP.

**Proof:** The corollary follows from Theorem 2, stating that \(R(\tilde{u}) = \tilde{R}(\tilde{u})\forall \tilde{u} \in \tilde{U}\), and from the fact that, since \(\tilde{U} = \tilde{U}\), the same policies apply to both the restricted and aggregated MDPs. \(\square\)

**Corollary 2:** By denoting with \(u^* \in \tilde{U}\) the optimal policy of the original MDP, with \(\tilde{u}^* \in \tilde{U}\) the optimal policy of the restricted/aggregated MDP under policy \(u^*\), and with \(u_G \in \tilde{U}\) the greedy policy, the following relation holds:

\[
R(u_G) \leq \tilde{R}(\tilde{u}^*) \leq R(u^*). 
\] (37)

**Proof:** Corollary 2 straightforwardly follows from Theorems 1 and 2, and from Corollaries 1 and 2. \(\square\)

**Corollary 3:** The policy \(u = (u_0, \ldots, u_{N-1}, u_{G,N}, \ldots, u_{G,M})\), obtained from the optimal restricted policy \(\tilde{u}^* = (\tilde{u}_0, \ldots, \tilde{u}_{N-1})\) computed by solving the aggregated MDP, is equal to the optimal policy \(u^*\) of the original MDP if it states that any connection must be accepted when \(\eta(t) \leq \eta_{LOW}\), i.e., if \((u_0, \ldots, u_M) = (u_{G,N}, \ldots, u_{G,M})\). Otherwise, \(u\) is a sub-optimal policy, which, in the worst case, returns the same expected reward of the greedy policy.

**Proof:** Corollary 3 is a consequence of Theorems 1 and 2, and of Corollaries 1 and 2. \(\square\)

Figure 3 shows an example of an original MDP together with its associated restricted and aggregated MDPs, with \(C = 2, (M + 1) = 7, (N + 1) = 5\).

### 2.3. Proposed Procedure to Approximate the Optimal Restricted Policy

The computation of the optimal restricted policy from the aggregate MDP improves the problem scalability,
depending on the percentage of aggregated states. However, to be sure to obtain the same optimal policy of the restricted MDP, the stationary probabilities of the states of the restricted MDP under the optimal restricted policy should be known in advance (see Eqs. (33) and (34)), but the computation of such probabilities requires the knowledge of the optimal restricted policy itself!

To overcome this problem, the proposed approach consists of four steps:

**Step 1** Computation of the stationary probabilities of the restricted MDP under greedy policy (which is known a priori): \( \pi_i(\bar{u}_G), i = 0, \ldots, N - 1 \).

**Step 2** Computation of the aggregate transition probabilities (33) and of the aggregate reward (34) under the restricted greedy policy \( \bar{u}_G \), i.e., of the aggregated MDP: \( \{ S, A, \bar{T}(\bar{u}_G), \bar{R}(\bar{u}_G) \} \).

**Step 3** Computation (via DP algorithms) of the optimal restricted policy \( \bar{u}^* = (\bar{u}_{0}, \ldots, \bar{u}_{N-1}) \) of the aggregated MDP \( \{ S, A, \bar{T}(\bar{u}_G), \bar{R}(\bar{u}_G) \} \).

**Step 4** Retrieving of the sub-optimal policy \( (\bar{u}_{0}, \ldots, \bar{u}_{N-1}, \bar{u}_{G,N}, \ldots, \bar{u}_{G,M}) \) of the original MDP.

**Remark 1:** The obtained policy \( \bar{u}^* \) is sub-optimal, since (1) it considers restricted policies only, and (2) it is based on approximated transition probabilities. The effectiveness of the proposed approach will be evaluated by numerical simulations in the Section 4 by considering both performances and complexity. The computation of the stationary probabilities under the greedy policy has a closed form (see [15]) and, thus, its computation cost is negligible. Also the space required to store the policy is reduced: the policy of the original MDP is a \((M + 1) \cdot C\) matrix, whereas the aggregated policy is a \((N + 1) \cdot C\) matrix.

**Remark 2:** Considering the LP problem formulation, the effect of reducing the policy space is the reduction of the number of variables: in fact, for each state \( x_i, i = N, \ldots, M \), the set of feasible policies \( A_i \) is reduced to the index \( k^* \) of greedy action \( a_k \) only; correspondingly, only one unknown \( y_{ik} \) is associated to each state \( x_i, i = N, \ldots, M \). The effect of the successive state aggregation is twofold: (i) the unknowns associated to the states \( x_i, i = N, \ldots, M \), are replaced by only one unknown; (ii) since the number of states decreases from \( M + 1 \) to \( N + 1 \), the number of problem constraints decreases from \( M + 1 \) to \( N + 1 \) as well. In conclusion, the policy reduction decreases the problem size (defined by the number of unknowns times the number of constraints) by reducing the number of unknowns, whereas the state aggregation further decreases the problem size both by reducing the number of unknowns and the number of constraints.

3. Connection Admission Control in UMTS Networks

In this paper, the UMTS network considered as an example of WCDMA network; the CAC function is
examined with respect to the problem of admitting new traffic in the uplink of an UMTS cell.

3.1. UMTS Networks

With the European third generation cellular system, UMTS, CAC becomes a very complex problem due to the “soft capacity” of WCDMA systems; the system is mainly interference-limited and a lot of services with different traffic models and quality requirements are supported [25,4,31]. The acceptance of a new connection depends on the SIR (Signal-to-Interference Ratio) values achievable by each existing connection once the new one is activated. These values relate to the emitted powers and, due to Power Control algorithm, depend on the location of mobile users. Since the power available at each Base Station is limited, the number of users that can be served is large if the Mobile Stations are close to the Base Station, and small if they are far away. The Power Control mechanism adopted by UMTS controls the power emitted on each channel in order to keep the SIR at a target value. In normal conditions, a stable state is reached after some iterations and all channels achieve the target SIR.

Two possible scenarios should be addressed upon the acceptance of a new connection:

1. the new connection is safely activated, since a new equilibrium can be achieved;
2. the new connection is erroneously admitted and a new stable state cannot be reached due to the levels of interference as well as to power constraints. The Power Control mechanism controls the power emitted on each channel in order to keep the SIR at the receiver at a target value. In normal conditions, a stable state is reached after some iterations and all channels achieve the target SIR.

Ideally, CAC should be able to accept a connection only if a new equilibrium of the power control can be reached and to reject it otherwise [29,16,32,8]. This ideal behaviour can be obtained with a complete knowledge of the propagation conditions and sources activity. More practical schemes implemented with a distributed control must cope with a limited knowledge of the system status and may erroneously accept or reject a connection.

3.2. UMTS Uplink Admission Control

UMTS CAC is based on the interference levels in the air interface. The received wideband interference power, $I_{total}$, can be divided into the powers of own-cell (intra-cell) users, $I_{own}$, other-cell (inter-cell) users, $I_{otb}$, and background and receiver noise, $P_N$:

$$I_{total} = I_{own} + I_{otb} + P_N.$$

The uplink noise rise, $NR$, is defined as the ratio of the total received power to the noise power:

$$NR = \frac{I_{total}}{P_N}. \quad (39)$$

Let us assume that, if a new connection is admitted, it will increase the interference by $\Delta I$; then, the new connection can be admitted if the new resulting total interference level is lower than the threshold value $I_{MAX}$, which is equal to the $NR$ and is set by radio network planning:

$$I_{total} + \Delta I \leq I_{MAX}. \quad (40)$$

To ease the estimation of $\Delta I$, the uplink load factor $\eta_{UL}$ is introduced, which. $\eta_{UL}$ is used as the uplink load indicator, is defined as follows:

$$\eta_{UL} = \frac{(NR-1)}{NR} = \frac{1-PN}{I_{total}}. \quad (41)$$

The uplink load factor $\eta_{UL}$ can be also calculated as the sum of the load factors of the on-going connections that are connected to the base station:

$$\eta_{UL} = (1 + i) \sum_{c}^{C} \left( N^{(c)} \Delta L^{(c)} \right), \quad (42)$$

where $i$ is the other-to-own cell interference ratio, $C$ is the number of classes (e.g., audio calls, video calls, data) supported by the network, $N^{(c)}$ is the number of on-going connections of class $c$, $\Delta L^{(c)}$ is the load increase—referred to as load factor—of a connection of class $c$.

The load factor is defined as follows [8]:

$$\Delta L^{(c)} = \frac{1}{1 + \nu^{(c)} R^{(c)}(E_b/N_0)^{(c)}}, \quad c = 1, \ldots, C, \quad (43)$$

where $W$ is the chip rate (3.84 Mchip/s), $R^{(c)}$ is the bit rate of class $c$, $(E_b/N_0)^{(c)}$ is the energy per user bit divided by the noise spectral density of class $c$, $\nu^{(c)}$ is the activity factor of class $c$ (which represents the fact that the traffic has a variable rate). For each class, the parameters allowing the CAC to compute the load factors are declared.

The statistics on the traffic generated by the users within a cell are usually given in terms of BHCA (Busy Hour Call Attempts), equal to the mean number of call attempts done by each user during the network traffic.
busiest hour, and \( MHT \) (Mean Holding Time), which is the average duration in minutes of a call. Arrival and termination rates are computed as follows:

\[
\begin{align*}
\lambda_c &= \frac{BHCA_c}{60} U \quad [1/\text{min}] \quad ; c = 1, \ldots, C, \\
\mu_c &= \frac{1}{MHT_c} \quad [1/\text{min}].
\end{align*}
\] (44)

where \( U \) is the number of users present in the cell.

Table 1 shows an example of declared parameters (with the resulting load factors) and of the traffic statistics for three classes.

Following the inter-cell mobility model considered in [4] and [33], connection handoffs (which happen when an active mobile terminal moves to a different cell) are considered as intra-cell connection attempts/terminations.

Since there is a relation between the interference and the load, similarly to Eq. (40) the new connection of a certain service class \( c \) is admitted if the current uplink load factor \( \eta_{UL} \) plus the load factor of the service class \( c \), \( \Delta L^{(c)} \), is lower than the load threshold \( \eta_{\text{MAX}} \):

\[
\eta_{UL} + \Delta L^{(c)} \leq \eta_{\text{MAX}}. \tag{45}
\]

The load threshold can be retrieved from experimental curves, measured by the UMTS operators, representing the relation between the interference and the load [8], as exemplified by Fig. 4. The availability of these curves allows one to model the admission control problem without being worried by the complex non-linear effects of the interference: through the use of the load factors, the interference-limited cell capacity is described by the load threshold \( \eta_{\text{MAX}} \).

### 3.3. Proposed CAC Policy Objectives and Update

Modern networks such as the UMTS one support a variety of applications (data, audio, video, and so on); such multi-service networks support traffic belonging to different classes of services, characterized by different requirements. Thus, connections belonging to different classes compete for bandwidth, and are regulated by the admission policy. Class-level requirements are usually specified in terms of blocking probabilities [12].

The operators’ objective is twofold: (a) on the one hand to efficiently exploit the network resources; (b) on the other hand to enforce fairness between classes by controlling the class blocking probabilities.

In this respect, if the blocking probabilities are not taken directly into account by the CAC policy, classes with low bandwidth requirements are greatly favoured [13]. The interest in the LP formulation is motivated by the possibility of adding class-level constraints. By defining the set \( B_c \) as the set of index \( k \) corresponding to the actions which blocks the admission of a class \( c \) call, the quantity

\[
P_c = \sum_{i=1}^{M} \sum_{k \in A_i} y_{ik}
\]

is the class \( c \) blocking probability.

The reward function of the LP (16) describes objective (a) only. To meet also objectives (b), in this paper we enforce fairness between classes by minimizing the squared differences between the class blocking probabilities, i.e., by adding a quadratic term to the reward function:

\[
R = w_{\text{load}} \sum_{i=0}^{M} \sum_{k \in A_i} r_i y_{ik} - w_{\text{fair}} \sum_{c=1}^{C-1} (P_c - P_{c+1})^2
\]

\[
= w_{\text{load}} \sum_{i=0}^{M} \sum_{k \in A_i} r_i y_{ik} - w_{\text{fair}} \sum_{c=1}^{C-1} \left( \sum_{i=0}^{M} \sum_{k \in A_i} y_{ik} - \sum_{i=0}^{M} \sum_{k \in B_{c+1}} y_{ik} \right)^2,
\]

\[
\] (47)
where the weights $w_{\text{fair}}$ and $w_{\text{load}}$ set the priority between fairness and expected load. Generic weighted fairness is obtained by properly setting weights multiplying the blocking probabilities. The obtained optimization problem is a Quadratic Programming (QP) one, whose solution cost is more demanding with respect to the LP formulation.

The stationary assumption of the MDP approach requires the CAC policy to be updated when traffic characteristics (such as arrival and termination rates) change. Network operators collect historical traffic statistics, which usually describe hourly traffic profiles [17]. By complementing these data with on-line measures, it is conceivable to update the admission policy on an hourly base [19]. The availability of interference measures might also suggest to update on-line the relation between interference and load and, thus, to update the policy even more frequently. In this respect, the availability of a fast method to compute the policy is crucial.

4. Simulation Results

This Section describes the numerical simulations obtained with CPLEX (ILOG 1996), a commercial optimization software package.

The three traffic classes shown by Table 1 were considered. The parameter setting of the two simulation runs is shown by Table 2, where $\eta_{\text{OFF}}$ is the offered load; by setting the number of users $U$, $\eta_{\text{OFF}}$ is computed as follows:

$$\eta_{\text{OFF}} = U \sum_{c=1}^{C} \left[ \Delta L^{(c)} \frac{ \text{BHCA}^{(c)} }{ 60 } MHT^{(c)} \right].$$  

(48)

The weights of the cost function were empirically set as specified in the table.

The first simulation is aimed at evaluating the effects of using the approximated policy (computed following the procedure of Section 2.3) with increasing values of $\eta_{\text{LOW}}$ (i.e., with increasing state aggregation). To evaluate the obtained fairness, we use the Jain’s fairness index (defined in [11]) which rates the fairness of the set of the $C$ values $P_c$, $c = 1, \ldots, C$. The result ranges from 0 (worst case) to 1 (best case): 

$$\text{fairnessindex} = \frac{ \left( \sum_{c=1}^{C} P_c \right)^2 }{ C \sum_{c=1}^{C} (P_c)^2}. $$

(49)

Note that if $\eta_{\text{LOW}}/\eta_{\text{MAX}} = 0$ the aggregated MDP is equal to the original MDP and the optimal policy $u^*$ is obtained, whereas if $\eta_{\text{LOW}}/\eta_{\text{MAX}} = 1$, the greedy policy $u_G$ is the only feasible policy of the aggregated MDPs.

Figure 5 shows the simulation results.

Figure 5a shows the fairness index vs. $\eta_{\text{LOW}}$. The fairness index remains above 0.98 until $\eta_{\text{LOW}} = 0.8$, whereas the fairness of the greedy policy $u_G$ is approximately 0.68.

The cost of enforcing fairness is limited, as shown by Fig. 5b, since the expected load is always more than 98% of the expected load obtained with $u_G$.

Note that the fairness index and the expected load obtained with the approximated policy are practically equal to the values obtained with the optimal policy $u^*$ until $\eta_{\text{LOW}} = 0.8 \eta_{\text{MAX}}$, and approach the values obtained with the greedy policy $u_G$ as $\eta_{\text{LOW}}$ approaches $\eta_{\text{MAX}}$.

Figure 5c shows how the QP problem size of the aggregated MDP decreases with $\eta_{\text{LOW}}$. Notably, the problem size with $\eta_{\text{LOW}} = 0.8$ $\eta_{\text{MAX}}$ is about one order of magnitude smaller with respect to the original problem size: $2.57 \cdot 10^6$.

Even more significant is Fig. 5d, which shows the central processing unit (CPU) time required to solve the QP on an Intel Mobile Centrino@1.73GHz: 922 sec. without aggregation, 15 sec. with $\eta_{\text{LOW}} = 0.8$ $\eta_{\text{MAX}}$. This result is important since it means that the proposed algorithm can be effectively used to compute on-line policies even in larger scenarios (with greater $\eta_{\text{MAX}}$ and/or with more classes).

To clarify the difference between a fair solution and the greedy one, Fig. 6 shows the blocking probabilities of the three classes obtained with the optimal policy, with the restricted optimal policy, with the aggregated policy, and with the greedy policy: while the first three policies obtain blocking probabilities between 15 and 20% for all the three classes (with fairness indexes equal to 0.9875, 0.9873 and 0.9842, respectively), the greedy policy obtains blocking probabilities ranging from 3 to 27% (with fairness index equal to 0.6858).

The second simulation is aimed at evaluating the effects of using a restricted policy and of using the approximated policy in different load conditions, with $\eta_{\text{LOW}} = 0.8 \eta_{\text{MAX}}$.

### Table 2. Simulation parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Simulation 1</th>
<th>Simulation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta_{\text{MAX}}$</td>
<td>0.29</td>
<td></td>
</tr>
<tr>
<td>$\eta_{\text{LOW}}/\eta_{\text{MAX}}$</td>
<td>{0.1, 0.15, \ldots, 1}</td>
<td>0.8</td>
</tr>
<tr>
<td>$\eta_{\text{OFF}}/\eta_{\text{MAX}}$</td>
<td>0.8</td>
<td>{0.5, 0.5, 0.6, \ldots, 1.1}</td>
</tr>
<tr>
<td>$\omega_{\text{load}}$</td>
<td>$1/\eta_{\text{MAX}}$</td>
<td></td>
</tr>
<tr>
<td>$\omega_{\text{fair}}$</td>
<td>$2.75 \eta_{\text{MAX}}/\eta_{\text{OFF}}$</td>
<td></td>
</tr>
</tbody>
</table>
Figure 7 shows the results. The figure highlights that, by setting $\eta_{\text{LOW}} = 0.8 \eta_{\text{MAX}}$, the restricted and the approximated policies obtain practically the same results of the optimal policy.

The results of Simulation 1 are confirmed at any load:

- The aggregate policy obtains fairness indexes between 97 and 99%, whereas the greedy policy obtains fairness indexes between 62 and 72%;
- The aggregate policy obtains expected loads between 97.5 and 99.5% of the expected load obtained with the greedy policy;
- The average CPU times required to compute the optimal policies and the aggregate policies were 911.3 and 14.7 sec, respectively.

5. Conclusions and Future Work

This paper proposes an Adaptive DP–based admission control algorithm. The algorithm consists of a
reduction of the policy space coupled with an approximated state aggregation approach, which leads to a dramatic reduction of the CPU time required to compute the admission policy. The algorithm was applied to the UMTS network scenario; simulation results showed that the proposed algorithm makes the implementation of the algorithm possible in a non-stationary environment, where the admission policy has to be updated periodically. The price to pay is that the obtained policy is only sub-optimal, even if the performance differences with the optimal policy are not significant in practice.

On-going work is focused on implementing the proposed algorithm in a real UMTS testbed provided by Polish Telecom R&D within the EU EuQoS project [6]; implementation requirements imply to address further problems related to the on-line estimation of the statistical characteristics of the different classes and of the relation between interference and load.

References