On Learning Bandwidth Allocation Models for Time-Varying Traffic in Flexible Optical Networks

Tania Panayiotou[1], Konstantinos Manousakis[1], Sotirios P. Chatzis[2], Georgios Ellinas[1]
[1] KIOS Research and Innovation Center of Excellence,
Department of Electrical and Computer Engineering, University of Cyprus
[2] Department of Electrical Engineering, Computer Engineering and Informatics,
Cyprus University of Technology,

***Abstract*—We examine the problem of bandwidth allocation (BA) on flexible optical networks in the presence of traffic demand uncertainty. We assume that the daily traffic demand is given in the form of distributions describing the traffic demand fluctuations within given time intervals. We wish to find a predictive BA (PBA) model that infers from these distributions the bandwidth that best fits the future traffic demand fluctuations. The problem is formulated as a Partially Observable Markov Decision Process and is solved by means of Dynamic Programming. The PBA model is compared to a number of benchmark BA models that naturally arise after the assumption of traffic demand uncertainty. For comparing all the BA models developed, a conventional routing and spectrum allocation heuristic is used adhering each time to the BA model followed. We show that for a network operating at its capacity crunch, the PBA model significantly outperforms the rest on the number of blocked connections and unserved bandwidth. Most importantly, the PBA model can be autonomously adapted upon significant traffic demand variations by continuously training the model as real-time traffic information arrives into the network.**

I. Introduction

With the emergence of new types of applications and services, the Internet traffic is exponentially growing [1]. Next generation optical networks are expected to support both the ever increasing traffic demand and the increased uncertainty in predicting the sources of this traffic. Over the last few years, and as the currently deployed optical networks are nearing a capacity crunch, they have undergone significant changes.

Flexible optical networks are considered today as a promising solution for coping with the increasing demand, due to their capability of efficiently utilizing the available spectrum resources [2]. Flexible optical networks are based on bandwidth variable transceivers (BVTs), a flexible grid, and network nodes that can adapt to the actual traffic needs [2]. In this type of networks, for establishing a connection, the Routing and Spectrum Assignment (RSA) problem must be solved. The routing (R) problem deals with finding a route for a source and destination pair. The spectrum allocation (SA) problem deals with allocating spectral resources to the routing path (the spectrum slots are occupied symmetrically around the nominal central frequency of the channel). The allocated spectrum must meet the slot continuity and contiguity constraints [3], subject to the constraint of no frequency overlap. Once a connection is established the spectrum width can be dynamically adapted (if feasible) in response to bandwidth variations. The RSA problem for time-varying traffic has been studied in [4]-[7] with the aim of best fitting the bandwidth requirements upon demand variations. A survey regarding the methods developed for the R problem can be found in [8], whereas regarding the SA problem, a number of SA policies have been developed that are in general categorized into fixed, semi-elastic, and elastic [5], [8].

In the *fixed* SA policies [4], [5] the allocated spectrum and the central frequency remain static for the entire lifetime of a connection. These policies lead to a sub-optimal use of the available resources as much of the allocated spectrum is most of the time wasted. In the *semi-elastic* SA policies [4], [5] the central frequency remains static but the allocated spectrum width can be expanded/reduced according to the actual bandwidth demand. The main difference with the fixed SA policies is that the unutilized slots can now be used for subsequent connection requests providing higher flexibility and better resource utilization. In the *elastic* SA policies [4]-[7], [9] both the allocated central frequency and the spectrum width can change. The spectrum width can be expanded/reduced according to the actual bandwidth demand and the central frequency can be shifted [5], [6], [9], [10]. The elastic SA policies offer better resource utilization but require the highest computational complexity and complex algorithms in the Path Computation Element for minimizing traffic interruptions if a reallocation policy is followed [5], [8]. Further, control plane extensions are still required for allowing dynamically adjusting both the allocated spectrum and the central frequency.

Most SA policies are based on daily Internet traffic patterns that can be known a priori due to the periodic behavior of Internet traffic [5]-[7], [9]. The traffic patterns include information regarding the estimated peak rate of each connection request for each time interval (usually 24-hour patterns). The estimated peak rates are used by the SA policy followed in order to allocate just enough bandwidth for each connection. For handling a situation where more bandwidth is eventually requested than the estimated one, the estimated peak rate is multiplied by a certain oversubscription ratio [4].

Motivated by the fact that the Internet traffic demand has been shown to follow the log-normal distribution [11], in this work, instead of assuming that the daily traffic patterns are given in the form of estimated peak rates, we assume that they are given in the form of distributions describing the traffic

978-3-903176-07-2

demand uncertainty (the mean and variance of the log-normal distribution are given). In this work, the assumption throughout is that the distribution describes the aggregate traffic resulting from multiple users. We wish to infer from these distributions a predictive bandwidth allocation (PBA) model that best fits the future bandwidth demands. In particular, we wish to find a bandwidth allocation (BA) model that is capable of predicting the number of spectrum slots that will best fit the traffic demand fluctuations of the next time interval. We have formulated the problem as a Partially Observable Markov Decision Process (POMDP) as POMDPs have been proven to be very effective for addressing planning domain problems with uncertainty [12]-[14]. For finding the PBA model, the POMDP is solved by means of dynamic programming. Note that the approach used for training the PBA model does not need to know the underlying traffic demand distributions. It can utilize real-time information for continuously adjusting the model upon variations on the traffic demand. The training procedure can be performed continuously offline, given that enough traffic information is available. Large amounts of traffic information can be easily collected by monitoring the traffic demand fluctuations within short time intervals. Nevertheless, given the fact that we do not have available real traffic information, in this work, we made the assumption that the traffic demand distributions are known. These distributions are used as traffic demand data generators for training and evaluating the effectiveness of the proposed PBA model.

We assume a network that is elastically reconfigured at the beginning of each time interval (24 hourly intervals). For each network reconfiguration, an RSA heuristic is executed offline. The SA must adhere to the BA model followed. A connection is blocked if a feasible route and SA cannot be found. Between network reconfigurations, the bandwidth for the established connections is semi-elastically expanded/reduced according to the fluctuations of the actual traffic demand. If the allocated bandwidth is higher or equal to the requested one, then the connection bandwidth is semi-elastically expanded/reduced or it remains unchanged. If the allocated bandwidth is less than the requested one, then some of the requested bandwidth remains unserved.

The PBA model is evaluated and compared to a number of benchmark BA models that naturally arise from the assumption of traffic demand uncertainty. Specifically, the PBA is compared to the Highest BA (HBA), to the Maximum Probability BA (MPBA), and to the Expected BA (EBA) models on a network that is operating at its capacity crunch.We show that the PBA model significantly outperforms the rest regarding the unserved bandwidth.

II. Bandwidth Allocation Models

We assume that the traffic demand is log-normally distributed [11] and that traffic demand information is available for a 24-hour period and for N source-destination pairs (connections). In particular, we assume that each connection is described by a set of traffic demand distributions, with each distribution describing the traffic demand fluctuations within a single time interval. In general, the log-normal distribution is asymmetrically distributed around its mean value and is suitable for describing data with heavy-tails and skewness.

The traffic demand fluctuations for each time interval $\{t\}_{t=1}^{24}$ and for each connection $\{n\}_{n=1}^{N}$ are described by $Z_{tn} \sim LN(\mu_{tn}, \sigma_{tn}^2)$. We assume that $z_{tn} \in (0, B)$ and that $B < B'$, where $z_{tn} \in Z_{tn}$, B is equal to the feasible rate of the BVTs, and B' is equal to the total link capacity (all network links occupy B' spectrum slots). For making the learning procedure of the PBA model computationally tractable, we have discretized the distributions according to specific rate intervals. Specifically, we have divided B into a intervals in such a way that the a^{th} interval is given by $B_a = [(a-1)k, ak]$, where $(a-1)k$ is the minimum rate of B_a, ak is the maximum rate of B_a, and $a = 1, 2, .., \frac{B}{k}$. Then we evaluated for each time interval t, for each connection n, and for each B_a, the probabilities $p_{tn}^a = P[z_{tn} \in B_a]$, where p_{tn}^a is the probability of connection n requesting at t a number of spectrum slots between $(a-1)k$ and ak. Since the traffic demand distributions are in this work randomly generated and may not be perfectly fitted to the tunability capabilities of the BVTs assumed, we have also evaluated $p_{tn}^0 = P[z_{tn} > B]$ to handle the distributions that generate rates above the feasible rate of the BVTs. By doing so, we managed to generate a valid discrete probability distribution. Without loss of generality, we assume that $B_0 = 0$ with probability p_{tn}^0.

For a network that is already configured and operating at t', a BA model indicates for each connection n the bandwidth allocation action a that must be taken for reconfiguring the network at the next time interval t. If the BA model indicates an action a for the connection n, then the number of spectrum slots Δ_{tn} that must be allocated to connection n are given by $\Delta_{tn} = \max\{B_a\}$. Note that the actions are actually the indices to the B_a intervals, and thus, for simplicity, the same notation is used for both the actions and the indices of the rate intervals. We assume that a network reconfiguration takes place at the beginning of each time interval t and is computed offline during the previous time interval t'. We now proceed with the description of the BA models developed.

1) Highest BA (HBA) Model: Indicates for each connection n and each upcoming time interval t, the BA action a that corresponds to the highest possible bandwidth demand. Specifically, $\Delta_{tn} = \text{argmax}_{a|p_{tn}^a>0}\{\max\{B_a\}|a = 0, 1, .., k\}$.

2) Maximum Probability BA (MPBA) Model: Indicates for each connection n and each upcoming time interval t, the BA action a that corresponds to the bandwidth interval with the maximum probability. Specifically, $\Delta_{tn} = \text{argmax}_a\{p_{tn}^a|a = 0, 1, .., k\}$.

3) Expected BA (EBA) Model: Indicates for each connection n and each upcoming time interval t, the BA action a that corresponds to the bandwidth interval in which the expected bandwidth of the distribution of interest belongs. Specifically, given that the expected bandwidth is $E[\Delta_{tn}] = \sum_{i=0}^{k} \max\{B_i\} p_{tn}^i$, then $\Delta_{tn} = \max\{B_a\}$, where $E[\Delta_{tn}] \in B_a$.

4) Predictive BA (PBA) Model: Our stochastic BA problem is formulated as a Partially Observable Markov Decision Process

(POMDP). POMDPs generalize Markov Decision Processes (MDPs) that are usually used in heuristic search and planning for accommodating stochastic actions and full state observability [15]. POMDPs differ from MDPs in that the states are not observable but are estimated from observations.

Formally, a POMDP is defined as a tuple $\{S, A, T, O, \Omega, b_0, R, \gamma\}$, where S is the set of states. A is the set of actions, $T(s'|s,a)$ defines the distribution over next state s' to which the agent may transition after taking action a from state s, O is the set of observations, $\Omega(o|s,a)$ is a distribution over observations o that may occur as a result of taking action a and entering state s, $R(s,a)$ is the reward function that specifies the immediate reward for taking action a at state s, $\gamma \in [0,1)$ is the discount factor that weighs the importance of current and future rewards, and b_0 is the vector of initial state distribution such that $b_0(s)$ denotes the probability of starting at state s.

In general, at each time step, the environment is at some state $s \in S$. The agent takes an action $a \in A$, and the environment transitions to state s' with probability distribution $T(s'|s,a)$. At the same time, the agent receives an observation $o \in O$ which is associated with the latent (unobservable) state s' according to some conditional likelihood function $\Omega(o|s',a)$. Finally, the agent receives a reward equal to $R(s,a)$. Then the process repeats. The goal is for the agent to choose actions at each time step t that maximize its expected future discounted reward $E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$.

In our BA problem, let us consider that the correlation between optimal network configuration and traffic demand patterns is not static, but may fluctuate on the grounds of longer term temporal dynamics. In that case, we must be capable of inferring these changes and adapting our policies accordingly. The essence of POMDPs addresses this consideration; POMDPs effect this goal by postulating that, at each time point, the modeled system has some latent state, s. Depending on the latent state, s, the same traffic demand requires a different policy of network reconfiguration, due to the different longer-term trends/dynamics that this latent state information encapsulates.

On this basis, for formulating the POMDP according to our BA problem, S, A, T, O, Ω, b_0 and R are now defined, for each connection n in the network, as follows:

- $S = \{s | s = 0, 1, ..., k\}$ with each state s representing the number of spectrum slots assigned to connection n.
- $A = \{a | a = 0, 1, ..., k\}$ with each action a representing the interval B_a, and hence the number of spectrum slots Δ_* that must be allocated to n.
- $T(s'|s,a)$ defines the probability of transitioning to state s' if action a is taken at s. Note that for each connection n we assume that a spectrum size transition is always possible (for simplicity a network with infinite capacity is assumed - the network capacity limitations are considered during the RSA algorithm in which the trained BA models are incorporated).
- $O = \{o | o = 0, 1, ..., k\}$ with each observation o representing the interval B_o in which the requested (observed) rate belongs.
- $\Omega_n(o|s,a) = p_{tn}^o$ is the observation distribution of connection n. The observation distribution generates at each time step t the true bandwidth demand of n.
- $R(s,a)$ is the reward function that specifies the immediate reward for taking action a at state s, and cannot be known a priori. The immediate reward for each state-action pair depends on what the agent observes at s' after action a is taken at s. On this basis, it is evaluated on the fly during the learning and exploration procedure of the POMDP (see Algorithm 1). For evaluating $R(s,a)$, we define instead a reward function $r(s', a, o)$. Each element of $r(s', a, o)$ specifies the reward received when o is observed at s', after action a is taken at s. Specifically,

$$r(s', a, o) = \begin{cases} -C, & \text{if } a < o \\ \exp[M(k - a + o)], & \text{otherwise} \end{cases} \quad (1)$$

Equation 1 indicates that if the requested demand (o) is higher than the allocated bandwidth (a), then the reward function r returns the constant negative reward $-C$, penalizing the action taken at s. On the other hand, if the requested demand (o) is lower than the allocated bandwidth (a), then a positive reward is received. According to Eq. 1 the positive reward is calculated as $\exp[M(k - o + a)]$, where M is a constant number, and returns a greater reward when the requested bandwidth is closer to the allocated one. Note that the reward is increasing exponentially as the requested bandwidth becomes closer to the allocated one, in order to allow the PBA model to learn the importance of allocating a bandwidth that is near the requested one. Equivalently, the PBA model is guided to avoid allocating at each time interval the highest possible bandwidth in an attempt to ensure a positive reward. By doing so, we aim at reducing both the unserved bandwidth as well as the unutilized allocated bandwidth (PBA is guided to strike a balance between the unserved bandwidth and the allocated one). Note that b_0 is set to $b_0(s) = \frac{1}{k} \ \forall s$ indicating that connection n can be initialized at any possible state s.

Commonly, POMDPs are solved by formulating them as completely observable MDPs over the *belief states* (posterior probability) of the agent [16]. Specifically, in POMDPs, as the true state is not observable, the agent must choose its actions based only on past actions and observations. Normally, the best action to take at time step t depends on the entire history of actions and observations that the agent has taken so far. However, the probability distribution over current states, known as the belief, is a sufficient statistic for a history of actions and observations [13]. In discrete state spaces, the belief state at step $t+1$ can be computed from the previous belief, b_t, the last action a, and observation o, by the following application of Bayes rule [13]

$$b_{t+1}^{a,o}(s) = \Omega(o|s,a) \sum_{s' \in S} T(s|s',a) b_t(s') / Pr(o|b,a), \quad (2)$$

where $Pr(o|b,a) = \sum_{s' \in S} \Omega(o|s',a) \sum_{s \in S} T(s'|s,a) b_t(s)$. The Bellman equation for the resulting belief MDP is [13]:

$$V_t^*(b) = \max_{a \in A} Q_t(b,a), \quad (3)$$

$$Q_t(b,a) = R(b,a) + \gamma \sum_{o \in O} Pr(o|b,a) V_t(b^{a,o}), \quad (4)$$

where the value function $V(b)$ is the expected discounted reward that an agent will receive if its current belief is b, $Q(b,a)$ is the value of taking action a at belief b, and $R(b,a)$ is the expected reward given by $\sum_{s \in S} R(s,a)b(s)$. As the exact solution of the Bellman equation (Eq. (3)) is intractable for large spaces [17], in this work, the Real-Time Dynamic Programming-Bel (RTDP-Bel) [18] heuristic algorithm is used for finding an optimal policy. In RTDP-Bel a greedy policy π_V is used for finding an optimal policy, where $\pi_V(b) = \text{argmax}_{a \in A} Q_t(b,a)$.

The RTDP-Bel is an asynchronous value iteration algorithm that converges to the optimal value function and policy over the relevant belief states without having to consider all the belief states in the problem. For achieving this, the RTDP-Bel uses an admissible heuristic function or lower bound h as the initial value function. Provided with such a lower bound, RTDP-Bel selects for update the belief over the states that are reachable from the initial state b_0 through the greedy policy π_V in a way that interleaves simulation and updates. For the implementation of the RTDP-Bel, the estimates $V(b)$ are stored in a hash table that initially contains only the heuristic value of the initial state, b_0. Then, when the value of a belief $b^{a,o}$ that is not in the table is needed, a new entry for $b^{a,o}$ with value $V(b^{a,o}) = h(b^{a,o})$ is allocated. These entries are updated following Eq. (3) when a move from s is performed. The RTDP-Bel algorithm is described analytically in [18].

In this work, the state-of-the-art RTDP-Bel algorithm is slightly modified to fit our problem formulation, incorporating the reward function defined in Eq. 1. The modified RTDP-Bel algorithm is described in Algorithm 1. Algorithm 1 is independently executed for each connection n in the network, and hence for each connection a different PBA model is evaluated. In Algorithm 1, an *episode* is defined as the sequence of actions and observations received for all the time intervals $\{t\}_{t=0}^{24}$. According to Algorithm 1, in each time interval t a single observation is sampled from $\Omega_n(o|s,a)$. It is true, however, that within t a number of traffic demand fluctuations may occur. The algorithm will eventually obtain enough observations and will converge to an optimal PBA through the iteration over a large number of episodes. In Algorithm 1 the target belief is at $t = 24$.

III. Routing and Spectrum Allocation

The RSA heuristic is executed for each time interval $\{t\}_{t=1}^{24}$ and for each connection $\{n\}_{n=1}^{N}$, during the previous time interval t'. Network reconfiguration takes place at the beginning of each time interval t. For each t, the RSA is solved without considering the network configuration at t' (complete connection reallocation is allowed). Specifically, for each t, the RSA finds a route and a spectrum allocation for each connection n, starting with the connection, n', requesting the maximum number of slots $\Delta_{tn'}$. For the R problem, the k-shortest path algorithm is used [19], while for the SA problem the first-fit algorithm is used, subject to the spectrum continuity, spectrum contiguity, and no frequency overlap constraints [3]. An ILP formulation was also developed for BA model evaluation, demonstrating that the proposed PBA model outperforms the benchmark BA models (omitted due to space limitations).

Algorithm 1 Modified RTDP-Bel alg. for each connection n

1: **Start** with $b = b_0$.
2: **Sample** state s from its probability distribution $b(s)$.
3: **Evaluate** each action a at belief state b as:

$$Q(b,a) = R(b,a) + \gamma \sum_{o \in O} Pr(o|b,a) V(b^{a,o}),$$

initializing $V(b^{a,o})$ to $h(b^{a,o})$ if $b^{a,o}$ is not in the hash.
4: **Select** action a that maximizes $Q(b,a)$.
5: **Update** $V(b)$ to $Q(b,a)$.
6: **Sample** next state s' from its probability distribution $T(s'|s,a)$.
7: **Sample** observation o from its probability distribution $\Omega_n(o|s',a)$
8: **Sample** reward r from the reward function $r(s',a,o)$
9: **Set** $R(s,a)$ equal to $r(s',a,o)$.
10: **Compute** $b^{a,o}$ using (2).
11: **Finish** if $b^{a,o}$ is target belief, else $b := b^{a,o}$, $s := s'$, and go to 3.

IV. Performance Evaluation

The performance of the BA models was evaluated and compared on the generic Deutsche Telekom (DT) network [4]. Each spectral slot in the network was set at 12.5GHz, with each fiber link utilizing $B' = 180$ slots. The feasible range of the BVTs was set to $B = 100$ slots. Note that this link capacity was chosen for reducing the computational time in our MATLAB machine with a CPU @2.60GHz and 8GB RAM. Bandwidth B was divided into $k = 10$ rate intervals $\{B_a\}_{a=0}^{k}$. Hence, each BA model can choose at each t and for each n amongst 11 spectrum allocation actions. Each action a indicates that $\Delta_{tn} = a \times k$ spectrum slots must be allocated at time interval t for connection n. Twenty-four time intervals were assumed.

In total 14 connection were considered, with seven of the connections following the log-normal distribution and the rest set to be static. The static connections were added as a simple approach for bringing the network at its capacity crunch and enabling the performance evaluation of the BA models on such a network. Regarding the stochastic connections, their traffic demand parameters, for each connection n and time interval t, are given by the $(\mu_{tn}, \sigma_{tn}^2)$ parameters of the log-normal distribution. The σ^2 parameters were uniformly generated in the range $[0,1]$ and the μ parameters were uniformly generated in the range $[0,5]$. Note that for simplicity, and without loss of generality, we did not consider that the mean rate value (μ) between sequential (in time) traffic distributions increases/decreases smoothly. Such a consideration would not affect the learning procedure or the efficiency of the PBA model. Regarding the static connections, their bandwidth demand Δ_* was set to be constant for all the time intervals. Δ_* values were randomly generated in the range $[20,60]$.

A. Training the PBA Model

For training the PBA model, the discount factor γ was set to 0.95 (typical value for POMDP training). Constants C and

M of the reward function (Eq. 1) were set to 10000 and 10, respectively. Note that a complete examination of how γ, C, and M values affect the trained PBA model could not be performed in this paper due to space limitations, and it is left for future work. A unique PBA model was trained for each one of the seven stochastic connections. For each PBA model, RTDB-Bel was iterated over 6000 episodes of learning, which interleaved simulation and model updates (the model was updated after every 20 simulated episodes). After each model update, 200 test episodes were generated with the model fixed, for evaluating the model's efficiency. For each test episode, the model returned the total reward, the total allocated bandwidth, and the total number of negative rewards received. These values were averaged over all 200 episodes.

Figures 1-3 illustrate how the average reward, the average allocated bandwidth, and the average number of negative rewards evolve over the training time of the PBA model. Training time is given in hours and corresponds to the time required for training and testing the model (for the 6000 episodes). A model update is indicated with a circle in Figs. 1-3 (250 total model updates). Figures 1-3 correspond to the PBA model of connection $n = 1$ (similar figures were obtained for all the other connections but are omitted due to space limitations). Figure 1 shows that the PBA model performs better as the training procedure evolves. The average reward increases with the number of model updates (training time) as the agent learns to take better bandwidth allocation decisions. Fewer negative rewards are received (Fig. 3) and the allocated bandwidth converges near the requested one (Fig. 2).

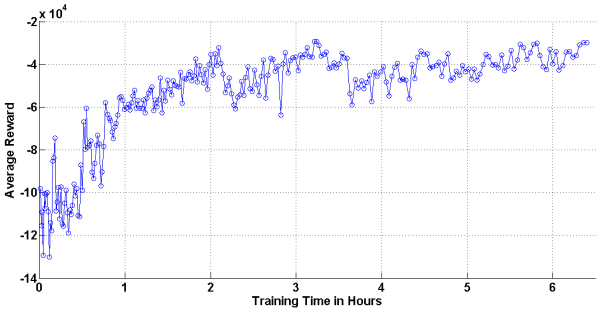

Fig. 1: Average reward over training time.

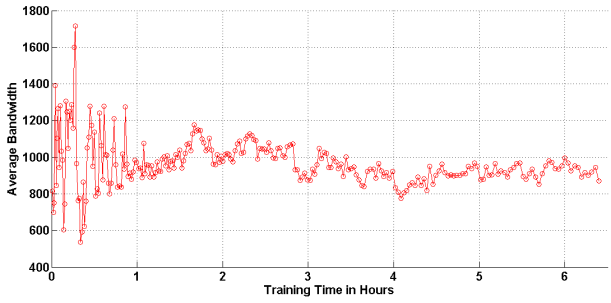

Fig. 2: Average allocated bandwidth over training time.

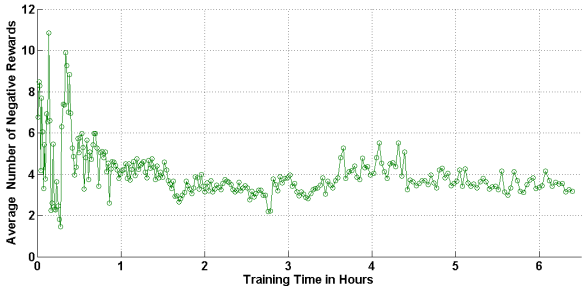

Fig. 3: Average negative rewards over training time.

In our simulations, each connection was trained for the same number of episodes and the last PBA model obtained was utilized for the network reconfigurations (during the RSA heuristic). Each model required up to 6 hours of training and testing. An action was generated within milliseconds from each model. Note that the models for each connection can be trained in parallel and independently from each other, and thus the number of time-varying connections does not affect the scalability of the PBA model. Further, the training procedure can be continuously performed for automatically adjusting the models upon significant variations on the traffic demand distributions; an important capability of the proposed method, given that the future traffic demand is expected to increase in uncertain ways (we cannot know the magnitude of a future traffic demand or the sources of this traffic).

Table I demonstrates how each trained PBA model performs against HBA, MPBA, and EBA. For each BA model we generated 200 episodes of actions and observations assuming a network with infinite capacity. The allocated bandwidth and the number of times an observation was greater than the action taken (negative reward) were averaged over these episodes. Note that a single observation was drawn for each action taken. Table I shows both the average allocated bandwidth and the average number of negative rewards.

TABLE I: BA Model Comparison

	Average Allocated Bandwidth				Average No. of Negative Rewards			
n	HBA	MPBA	EBA	PBA	HBA	MPBA	EBA	PBA
1	1490	560	690	868	0	4.77	3.3	3.1
2	1730	500	470	1171	0	7.2	4.2	2.1
3	1310	370	480	950	0	3.48	2.5	0.9
4	1400	370	580	750	0	6	3.3	4.5
5	1690	350	488	665	0	6.13	4	3.5
6	1420	320	520	830	0	6.7	4.3	4.1
7	1700	380	480	667	0	6.1	4.1	3.4

According to Table I, PBA tends to allocate fewer slots compared to HPBA and more slots compared to MPBA and EBA. Hence, PBA increases the negative rewards received compared to HBA that never receives a negative reward. MPBA and EBA receive on the average more negative rewards than PBA as they tend to allocate fewer slots than PBA. This is a consequence of the reward function (Eq. 1) defined for PBA training that aims at allocating at each time interval a bandwidth that is close to the requested one.

B. Network Performance Evaluation

The RSA algorithm was solved on the DT network for each BA model and each time interval t. For each t, an action was generated for each connection n and RSA was solved having as inputs the rates Δ_{tn} indicated by the model's actions. RSA required at most 15 seconds for finding a feasible solution for each time interval. Between network reconfigurations the traffic demand fluctuated according to the given set of traffic demand distributions. For the traffic demand fluctuations we have drawn from each Z_{tn} the samples $\{z_{tn}^i\}_{i=1}^{60}$ representing the traffic demand fluctuations every minute of the hour. Sample $\delta_{tn}^i = z_{tn}^i$ denotes that connection n requests δ_{tn}^i spectrum slots at the i^{th} minute of time interval t.

For each established connection, the allocated Δ_{tn} slots were compared to each δ_{tn}^i in order to calculate the unserved slots and the excess (unutilized) allocated slots. The unserved slots for each episode are given by $U = \frac{1}{60\times 24}\sum_t\sum_n\sum_i|\Delta_{tn} - \delta_{tn}^i|$, if $\Delta_{tn} < \delta_{tn}^i$. The excess slots for each episode are given by $E = \frac{1}{60\times 24}\sum_t\sum_n\sum_i(\Delta_{st} - \delta_{tn}^i)$, if $\Delta_{tn} > \delta_{tn}^i$. Two-hundred episodes were generated for each BA model and the unserved and excess slots were averaged over these episodes. Table II shows the average number of unserved ($\bar{U}$) and excess ($\bar{E}$) slots per time interval. It also shows the average number of blocked connections ($\bar{\Pi}$) per episode.

TABLE II: BA Model Comparison on DT Network

	HBA	**MPBA**	**EBA**	**PBA**
Av.# of Excess Slots (E)	337	35.3	82	155
Av.# of Unserved Slots (U)	23	49	48	20.3
Av.# of Blocked Connections ($\bar{\Pi}$)	16	0	0	0

According to Table II, as expected, HBA allocates on the average a higher number of excess slots (337) compared to the other models. The high number of excess slots led, on the average, to 16 blocked connections (these connections are entirely terminated, each for an hour during a day). HBA is clearly not a feasible solution for a network operating at its capacity crunch. If we assume that the end user behavior remains the same during the unavailability period, the 16 blocked connection lead to 23 unserved slots (greatly unbalanced between the connections). Under this consideration, PBA outperforms HBA by 11%.

Table II shows that MPBA, EBA, and PBA significantly reduce the average excess slots by 80%, 75%, and 54%, respectively, compared to HBA. Consequently, these models, unlike HBA, did not cause any blocking. However, the traffic demand fluctuations within each time interval resulted in some unserved slots. In particular, PBA results on the average in 20.3 unserved slots, while MPBA and EBA, result on the average in 49 and 48 unserved slots, respectively. Hence, PBA outperforms MPBA and EBA, in terms of unserved slots, by approximately 58%. Overall, PBA predicts a bandwidth that more efficiently handles traffic demand fluctuations.

V. Conclusion

We proposed an effective formulation of a state-of-the-art POMDP method that learns by means of DP an optimal predictive BA model from a given set of traffic demand distributions that is consequently used for bandwidth allocation decisions during network reconfigurations. PBA is compared to the naturally arising HBA, MPBA, and EBA techniques and it is shown that it outperform HBA on the number of blocked connections, as well as MPBA and EBA on the unserved bandwidth that may occur during traffic demand fluctuations.

Acknowledgment

This work has been supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE) and from the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development. This work has also received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA Grant Agreement no. 630853.

References

[1] Cisco white paper, "The Zettabyte Era: Trends and Analysis," 2017.
[2] O. Gerstel, et al., "Elastic Optical Networking: A New Dawn for the Optical Layer?," *IEEE Comm. Mag.*, 50(2):s12–s20, 2012.
[3] K. Christodoulopoulos, et al., "Elastic Bandwidth Allocation in Flexible OFDM-Based Optical Networks," *IEEE/OSA J. Lightwv. Techn.* 29(9):1354–1366, 2011.
[4] K. Christodoulopoulos, et al., "Time-Varying Spectrum Allocation Policies and Blocking Analysis in Flexible Optical Networks," *IEEE J. on Selected Areas in Comm.*, 31(1):13–25, 2013.
[5] M. Klinkowski, et al., "Elastic Spectrum Allocation for Time-Varying Traffic in FlexGrid Optical Networks," *IEEE J. on Selected Areas in Comm.*, 31(1):26–38, 2013.
[6] S. Shakya, et al., "Spectrum Allocation for Time-varying Traffic in Elastic Optical Networks using Traffic Pattern," *Proc. OFC*, 2014.
[7] G. Shen, et al., "Maximizing Time-dependent Spectrum Sharing between Neighbouring Channels in CO-OFDM Optical Networks," *Proc. ICTON*, 2011.
[8] B. C. Chatterjee, et al., "Routing and Spectrum Allocation in Elastic Optical Networks: A Tutorial," *IEEE Comm. Surveys & Tutorials*, 17(3):1776–1800, 2015.
[9] K. Christodoulopoulos, et al., "Dynamic Bandwidth Allocation in Flexible OFDM-based Networks," *Proc. OFC*, 2011.
[10] F. Cugini, et al., "Push-Pull Defragmentation Without Traffic Disruption in Flexible Grid Optical Networks," *IEEE/OSA J. Lightwv. Techn.*, 31(1):125–133, 2013.
[11] I. Antoniou, et al., "On the Log-normal Distribution of Network Traffic," *Physica D: Nonlinear Phenomena*, 167(1–2):72–85, 2002.
[12] L.P. Kaelbling, et al., "Planning and Acting in Partially Observable Stochastic Domains," *Art. Intell. Journal*, 101(1–2):99–134, 1998.
[13] F.D.-Velez, "The Infinite Partially Observable Markov Decision Process," *Advances in Neural Information Proc. Systems 22*, 2009.
[14] S.P. Chatzis, D. Kosmopoulos, "A Non-stationary Infinite Partially-Observable Markov Decision Process," *Proc. ICANN*, 2014.
[15] D. Bertsekas, "Dynamic Programming and Optimal Control, (2Vols)". *Athena Scient.*, 1995.
[16] E. Sondik, "The Optimal Control of Partially Observable Markov Decision Processes over the Infinite Horizon: Discounted Costs", *Oper. Res.*, 26(2):282–304, 1978.
[17] C. Papadimitriou and J. Tsisiklis, "The Complexity of Markov Decision Processes," *Math. of Operations Research*, 12(3):441–450, 1987.
[18] B. Bonet, H. Geffner, "Solving POMDPs: RTDP-Bel vs. Point-based Algorithms," *Proc. IJCAI*, 2009.
[19] J.Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, 17(11):712-716, 1971.