Contextual Bandit for Cognitive Management of Self-Organizing Networks

1st Tony Daher
Orange Labs
Chatillon, France
tony.daher@orange.com

2nd Sana Ben Jemaa
Orange Labs
Chatillon, France
sana.benjemaa@orange.com

***Abstract*—Self-Organizing Networks (SON) concept is a technology that aims to improve the management and operation of mobile networks, through automatic configuration of network parameters. Even though SON functions are able to change network parameters automatically, the algorithms that run inside these functions still rely on parameters and rules that are manually defined by the operator, depending on its objectives. Thus, in order to realize a network that is self-organized as a whole, there is a clear need for a higher-level management entity that automatically translates operator objectives into SON configurations. In previous works, we have already studied and proposed an intelligent integrated management solution empowered with Reinforcement Learning (RL), namely the Cognitive Policy Based SON Management (C-PBSM).The C-PBSM is able to learn optimal SON configurations through direct interaction with the network. In this paper, we address crucial aspects of the mentioned approach, namely adaptability with different and varying network environments, transferability of the knowledge and the speed of convergence. We argue that the C-PBSM has major limitations with respect to these aspects. We consequently propose a context aware C-PBSM show that it is able to overcome the limitations of the C-PBSM.**

***Index Terms*—Radio Access Networks, Self-Organizing Networks, Reinforcement Learning, Policy Based Management.**

I. Introduction

The management of mobile networks has always been a challenging task for network operators, especially with the need to reduce the operational expenses while maintaining a good Quality of Service (QoS) and Quality of Experience (QoE) for the users, at a competitive price. Improving the network management efficiency has become even more crucial in the recent year, as the networks became more complex and the traffic demands increased at a high pace [1].

In this sense, autonomic solutions are already being deployed in today's networks, and will be the bedrock of 5G [2]. A first step towards autonomic networks was achieved with the 3GPP introduction of the Self-Organizing Networks (SON) functions in its release 8 [3]. A SON function operates as follows: it continuously receives measurements feedback from the network, and changes certain network parameters according to its algorithm [4]. Different SON functions can be deployed in the network to replace specific operational tasks such as coverage optimization or load balancing. Such a network is SON enabled, but cannot be considered as a self-organized network. A self organized-network takes as inputs the target key performance indicators (KPIs), translates them automatically into appropriate network parameters so that these high level objectives are fulfilled. For a network that is SON enabled, this can be done through managing properly all the SON functions in the network, to make them fulfill together the operator objectives, as shown in Figure 1.

978-3-903176-32-4 © 2021 IFIP

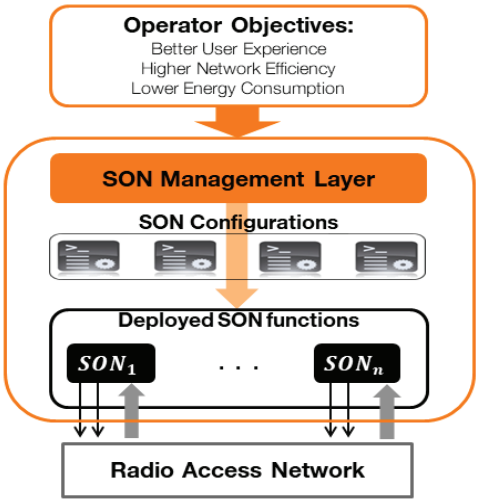


Fig. 1: PBSM Scheme

Figure 1 represents an integrated SON management framework, known as Policy Based SON Management (PBSM): the operator objectives are given as input to the PBSM, which automatically outputs the best identified policy. A policy is the set of SON configuration values (SCV). We consider that the operator objectives are expressed as target KPIs to be optimized in the network.

An approach for PBSM was first studied in the framework of the Semafour European project [5]. The approach is mainly based on pre-simulated SON Function Models (SFM) that map KPIs to SCV sets [6]. Even though in following works the generation of SFM was enhanced with machine learning [7], the proposed approach still relies strongly on input models, which might often differ from the network field reality.These models are static and do not reflect the dynamics of the radio environment. In [8], [9], the authors propose a

Cognitive-PBSM (C-PBSM), where a Reinforcement Learning (RL), specifically the stochastic Multi-Armed Bandit (MAB) algorithm, was used to empower the system with cognitive capabilities: learning through interaction with the real network.

The most straight forward formulation of the MAB is the following: there are K possible actions, known as arms, each having an unknown distribution of bounded rewards. At each iteration, the learning agent chooses an action, then only the reward of this action is revealed. The objective is to find as fast as possible, and with enough confidence, the optimal arms. It is hence a suitable and promising approach for online learning problems, where the policy is learned through interaction with the real environment

In the recent years, the MAB has seen considerable theoretical advances. It has also been used to enhance many control processes with cognition and intelligence, including specific SON functions. For instance in [11], [12], [13], [14], the authors propose self-optimization and self-healing SON functions with learning capabilities. On the other hand, in [15], [16] MAB algorithms show to be very effective in dealing with spectrum management and resource allocation problems.

In the previous works on C-PBSM, the convergence of the proposed algorithms is still far from being acceptable in real networks (several days to converge), knowing that the performance can be degraded during the exploration phase. Besides, Radio Access Networks (RAN) environments are very diverse and heterogeneous. They can be rural, urban, dense, have different types of required services or traffic profiles etc. The best policy learned in a certain section of the network is not guaranteed to still be the best one on other sections. The policies must be adapted to the network context. Finally, previous studies assume a stationary traffic, while it varies during the day [17]. The C-PBSM has to adapt its decisions to traffic variations.

We propose in this work a novel architecture and learning framework for the C-PBSM, based on contextual MAB, that we refer to as context aware C-PBSM. A single learning agent learns simultaneously over several small network clusters, geo-localized in different parts of the network, and finds the optimal policy for each network context. The contribution of this paper can be summarized as follows:

- The proposed algorithm finds optimal policies over different network contexts, including traffic variations.
- Learnt policies can be transferred to untrained sections of the network.
- Collaborative learning improves the convergence speed.

The remainder of this paper is organized as follows. Section II describes the proposed context aware C-PBSM. In section III we present the RL and MAB framework. We focus on the Bandit Forest contextual MAB algorithm that we consider for this work and we motivate our choice. Section IV presents a use case scenario and evaluates the performances of our approach. The last section concludes the paper and presents future works.

II. Context Aware C-PBSM

We consider a SON enabled network, with distributed SON functions i.e. each SON function is distributed through several instances on different cells. Each SON instance has its own control loop: it collects local KPI measurements and changes certain parameters depending on its objective, algorithm, and configuration (SCV set). On top of the SON layer, we consider an RL learning agent that learns and enforces the optimal SON configuration policy, according to the objectives specified by the operator.

The proposed functional architecture is depicted in figure 2. A learning cluster is typically a small group of neighboring cells, having similar characteristics, where the operator allows the testing of SON configurations and parameters. The cluster characteristics include the geographical area (rural, urban, railway, etc.), the technology (2G, 3G, 4G, etc.), the topology (n layer heterogeneous network) or the type of network equipment, and traffic characteristics. Note that several clusters, with given traffic characteristics, may correspond to the same network contexts, and their optimal policies are then learnt collaboratively.

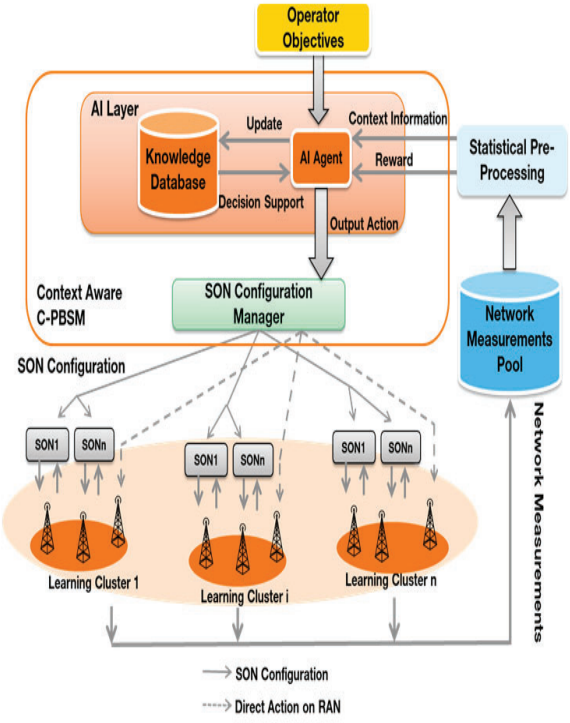


Fig. 2: Context Aware Distributed C-PBSM

The context aware C-PBSM consists of the AI Layer and the SON Configuration Manager:

- AI Layer: This layer contains 2 blocks: the AI Agent and the Knowledge Database. The AI agent is continuously interacting with the learning clusters. That is, it receives the reward and context information from each of the clusters, and chooses an action to be applied in each

of them. Then based on the cluster characteristics and additional context information, the AI agent deduces the appropriate network context. The action is chosen based on the knowledge acquired by the learning agent from previous iterations for this network context.
- The knowledge database is updated at each iteration. Note that the stored policies correspond to network contexts and not to each specific learning cluster. The more the clusters are diverse, the better they cover the context-action space of the network, hence making the Knowledge Database more complete and transferable to other untrained sections of the network. Moreover, increasing the number of learning clusters results in a faster convergence as much more SCV sets combinations are tested in parallel.
- The measured KPIs are kept in a registry called Network Measurements Pool. Before being forwarded to the AI Agent, the raw KPI measurements are statistically processed (typically averaging, smoothing, scalarization, etc.).
- SON Configuration Manager: This block receives the actions from the AI Agent and is in charge of enforcing the SCV set in the corresponding SON function instances in each of the learning clusters. It can also directly change certain network.

III. Context Aware C-PBSM: Implementation Based on Contextual Multi-Armed Bandit

MAB algorithms focus largely on optimizing the trade-off between exploring the environment and exploiting the acquired knowledge during the learning phase [18]. The contextual MAB is in its turn an extension of the MAB framework, that is able to observe context changes in the environment, and chooses the actions according to both the perceived rewards and the context information [19]. Contextual MAB algorithms have shown to be very useful in many real life problems such as recommender systems and web advertising [20].

In this paper we implement the random forest algorithm for the contextual bandit problem [21], that we henceforth refer to as Bandit Forest (BF). In most real life problems, the perceived reward depends on the taken action but also on the given context of the environment. A straightforward approach to deal with changing contexts would be to consider a stochastic MAB process per context, hence finding an optimal policy per context. However, such an approach would require a lot of time to converge because each MAB process would be updated only when its corresponding context is observed. Also when faced with a large number of contexts, then this approach would simply be infeasible. Instead, contextual MAB deals more efficiently with this problem by assuming that the context observations are not completely independent from each other. In other words, the rewards generated by choosing an arm a, under context c, can carry some information about the rewards perceived by choosing the same arm a, under a different context c'. We can find different form of context dependence assumptions in the literature [22], [23], [24].

The BF algorithm we implement belongs in its turn to the decision trees family. It considers that there is a subset of context variables that are more relevant than the rest. It hence reduces its exploration space by considering this subset of variables, and discarding the others. The BF algorithm was proposed and analyzed in detail in [21]. The authors studied the optimality of the algorithm in terms of sample complexity [25], where BF was shown to be optimal up to a logarithmic factor [21]. Furthermore, the dependence of the algorithm's sample complexity with the number of context variables is logarithmic, which means that the algorithm scales well with the number of context variables. This is a very important factor, because the complexity, heterogeneity and high dynamics of the RAN environment may require a huge number of context variables to describe a network section. The computational cost is as well linear with the time horizon and the number of context variables, allowing to process large sets of variables. All these characteristics make the BF algorithm well suited for real applications, notably for the context aware C-PBSM.

A. Contextual MAB

The contextual MAB problem is formalized as follows. Let A and K be respectively the set and the number of possible arms. Let V be the set of context variables and M their number. $\mathbf{r_t} \in [0,1]^K$ is the vector of bounded rewards and $\mathbf{x_t} \in \{0,1\}^M$ the binary context vector at iteration t. $D_{x,r}$ is the joint distribution on $(\mathbf{x}, \mathbf{r})$. Let $\pi : \{0,1\}^M \rightarrow A$ be a certain policy and Π be the set of policies. The objective of contextual MAB algorithms is to find the optimal policy π^*, that is the policy that maximizes the expected gain with respect to the distribution $D_{x,r}$:

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmax}} \mathbb{E}_{D_{x,r}}[r_{\pi(\mathbf{x}_t)}] \tag{1}$$

The learning agent learns the optimal policy in a sequential manner as explained in the following:

Algorithm 1 Contextual MAB

for each iteration t
- Agent receives context information $\mathbf{x_t}$
- Agent plays arm c_t according to policy $\pi(\mathbf{x_t})$
- Reward r_{t,c_t} is perceived according to $D_{x,r}$
- Agent updates policy π_t

The algorithm identifies the optimal policy based on the knowledge it has about the context-reward distribution $D_{x,r}$. To build this knowledge, the agent has to interact with the environment by exploring different actions and policies. The more the agent explores, the more the knowledge it has about $D_{x,r}$ is reliable, and so is its optimal policy. However, exploration leads to sub-optimal decisions of the agent, and hence sub-optimal rewards. Whence the necessity to balance exploration and exploitation. The MAB algorithm should find

the optimal policy while minimizing the expected cumulative perceived regret. defined as:

$$\mathbb{E}_{D_{x,r}}[R_n] = \sum_{t=0}^{n} \mathbb{E}_{D_{x,r}}[r_{t,c^*{}_t} - r_{t,c_t}] \quad (2)$$

R_n is the cumulative regret after n plays and $c_t^* = \pi^*(\mathbf{x_t})$ is the action chosen by the optimal policy.

B. Random Forest for the Contextual MAB

The BF algorithm is based on decision trees. A decision tree can be seen as a combination of rules, where only one rule is selected for a given input vector, which is in this case the context vector. Finding the optimal tree structure is NP-hard [21]. Instead, a greedy approach can be used to grow the decision tree, based on decision stumps (a decision stump is a one node decision tree) [26], [27].

A decision stump takes decisions based on the observation of one context variable. To maximize the perceived reward, the decision stump should identify the best context variable. That is the variable that maximizes, when observed, the expected reward of the best action for each of its values. After identifying the best context variable, the decision stump identifies the best action while observing the best context variable. In the BF algorithm, decision trees are grown by recursively stacking decision stumps, which means that a decision tree takes its decisions based on the observation of a subset of the best context variables, which is indeed a stronger learning model than the decision stump.

Moreover, the random forest improves the decision model by growing more than one tree, and by adding randomness in the process. That is, instead of searching for the most important context variables while splitting a node, each of the trees searches for the best context variable among a different random subset of the variables. This results in a wide diversity that generally results in a better model and improves the optimality of the decisions [28]. The BF algorithm bases thus its decisions on the output of the random forest.

We will not expose in this paper the details of the construction of the decision trees as it is out of the scope of this work. For additional information, please refer to [21]. In the next section we describe the system model and evaluate the performances of the proposed approach and compare them with a C-PBSM approach based on stochastic MAB.

IV. System Model and Simulation Scenario

A. Model Description

Consider a set Λ of network clusters distributed in different locations of the network. In each cluster $\lambda \in \Lambda$, we consider a set of deployed SON functions U_λ. N_u^λ is the number of instances of SON function u in sector λ ($u \in U_\lambda$). $N_u^\lambda = 0$ if $u \notin U_\lambda$. Each SON function has a set of SCV sets denoted as $C_u, \forall u \in U_\lambda, \forall \lambda \in \Lambda$. The set of possible SCV sets combinations in a network cluster λ is then defined as $A_\lambda = (C_{u_1})^{N_{u_1}^\lambda} \times (C_{u_2})^{N_{u_2}^\lambda} \times ... \times (C_{u_{|U|}})^{N_{u_{|U|}}^\lambda}$ where $u_1, u_2, ..., u_{|U|} \in U_\lambda$. The reward perceived by the learning agent for an action a at iteration t is assumed to be a linear combination of perceived KPIs Z_i and weights w_i, reflecting the operator's priority to maximize the corresponding KPI target:

$$r_a(t) = \sum_{i=1} Z_i w_i \quad (3)$$

At each iteration t, each cluster λ reports to the AI Agent a numerical reward value, depending on the reward definition in equation (3), and a feature vector $\mathbf{x_t^\lambda}$ carrying the context information at iteration t of cluster λ as described in figure 2. Note that the feature vector depends on both the network cluster λ and the iteration t. In fact, the feature vector carries information about both the topological and technological aspects of the access network in each cluster (e.g. environment, technology, information about the vendor's hardware, multi-layer or not, etc.) and time dependent network information (traffic information, types of services, special event etc.). At each reporting of the clusters, the AI agent updates the knowledge database, and outputs new actions to be applied in the clusters. In our case, as the AI agent runs the BF algorithm, the knowledge database stores the binary decision trees.

B. Simulation Scenario

We consider four network clusters for the learning process. Each cluster is composed of a macro cell and the first tier neighboring cells as represented in figure 3. Small cells can be deployed in a macro cell's coverage region, to serve a traffic hot-spot inside the coverage region of the macro cell. In this case the macro cell and small cell are referred to as master cell and slave cell respectively. We consider 3 SON functions

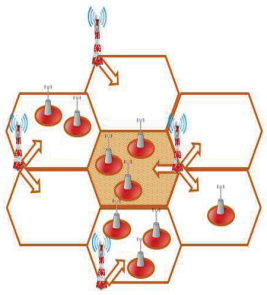


Fig. 3: Example of a Network Learning Cluster

in each cluster λ, each deployed in several instances according to the following:

a) Mobility Load Balancing (MLB): Deployed on each macro cell. Its objective is to balance the traffic load between the macro cells by tuning the Cell Individual Offset (CIO) of macro cells.

b) Cell Range Expansion (CRE): Deployed on each small cell, it tunes the CIO of the small cells to balance the load between small cells and the master macro cell.

c) Enhanced Inter Cell Interference Coordination (eICIC): Deployed on macro cells with small cells in their coverage area. eICIC manages Almost Blank Subframes (ABSF) [4] transmissions of macro cells in order to protect small cell edge users from macro downlink interference.

We further consider that the AI Agent can also directly act on certain network parameters. For example in this scenario we consider that it can turn on or off small cells through a sleep mode process in order to optimize the energy efficiency in the network. When a small cell enters sleep mode, the user equipment served by the small cell are handed over to the master macro cell.

For each iteration t and action a (SCV sets combination), the following KPIs, considered to be the most relevant in our scenario, are defined as:

- $l_{i,a}(t)$ is the load of cell i in network section λ
- $\bar{l}_a(t)$ is the average load in the considered section
- $\sigma_a(t) = \frac{\sum_{i=0}^{|\lambda|} l_{i,a}(t) - \bar{l}_a(t))^2}{|\lambda|}$ the load variance in cluster λ. $|\lambda|$ is the number of cells in cluster λ.
- $\overline{T}_a(t)$ is the average user throughput in the central macro cell
- $\overline{T'}_a(t)$ is the average small cell edge user throughput in the central macro cell coverage area
- $\overline{P}_a(t)$ is the average power consumption of the small cells deployed in the central macro cell coverage area
- $\sigma'_a(t)$ is the average load variance in the central macro cell and its slave small cells.

The perceived reward for a certain action a at iteration t is hence:

$$r_a(t) = w_1(1 - \sigma_a(t)) + w_2\overline{T}_a(t) + w_3\overline{T'}_a(t) + w_4\overline{P}_a(t) + w_5(1 - \sigma'_a(t)) \quad (4)$$

All the KPIs are normalized between 0 and 1.

For the action space, we consider the following SCV sets for each of the considered SON functions. The SCV sets differ in terms of activation threshold as well as the parameter ranges. We consider that all the instances of the same SON function are configured with the same SCV set in a given network cluster. We further consider that when Sleep Mode is activated (i.e. small cells are turned off), CRE and eICIC functions are turned off. This leaves us with 28 possible actions or arms.

MLB	**CRE**	**eICIC**	**Sleep Mode**
Off	SCV1	Off	Off
SCV1	SCV2	SCV1	On
SCV2	SCV3		
SCV3			

TABLE I

The vector of features describing the context reports to the AI agent information about: Macro inter-site distance, traffic and each of the neighboring cells, traffic in the small cells of the central cell, how many cell layers are there in the cluster (0 if homogeneous macro deployment, 1 if heterogeneous 2 layer network). We consider that the traffic in the cells can be Low, Medium, High or Very High.

V. Simulation Results

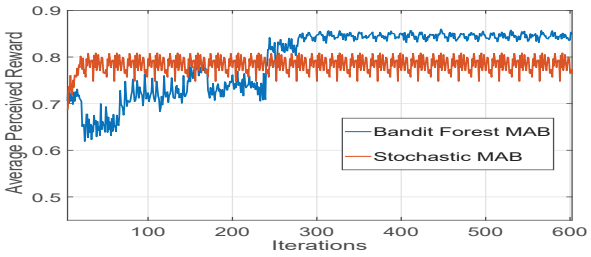
In this section we present the simulation results of the previously described scenario. We compare the performances of the contextual BF algorithm with a C-PBSM based on a stochastic MAB algorithm, namely the Successive Elimination (SE) algorithm [29]. Note that both BF and SE algorithms are learning simultaneously over the four considered training clusters.

For the simulation scenario, we consider that all the clusters are in urban areas and are heterogeneous two layer networks. We consider traffic changes in the different cells of the clusters, with four traffic levels: low, medium, high and very high. We also assume that the traffic is piece-wise stationary, that is the traffic variations that may occur during a RL iteration (which is 20 min in our case) are considered to be stationary and do not impact the stationary state of the system during this time interval. This is a reasonable assumption regarding the traffic profiles that can be found in the network. We run the learning process for different operator objectives.

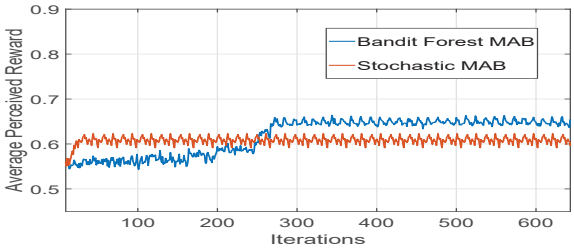
The average perceived reward is plotted in figure 4. The stochastic MAB is not able to observe the context changes. It estimates the average perceived reward of each arm regardless of the context changes. It identifies hence an action that has the highest empirical average over all the contexts. The contextual BF algorithm on the other hand constructs its policy by observing the context. Eventually, the BF algorithm identifies an optimal action for each of the observed contexts, performing better than the single action policy of the stochastic MAB as shown in figure 4 (w_i is the operator objective weight). Note that the different values of the rewards as well as their variance vary with the operator objectives. These changes are the consequences of the normalization of different KPIs with different distributions, ranges and behaviors, and do not reflect the optimality of the policy.

In figure 5 we plot the average perceived reward per context for a set of observed contexts in this scenario, for the previously considered objectives. We can see that the contextual BF algorithm performs always at least as good as the stochastic MAB. In other words, this means that the AI agent adapts the SCV sets of the SON functions, according to the observed context. This makes the contextual MAB more suitable for scenarios with context changes, with different optimal actions for different contexts, which is the case in real networks.

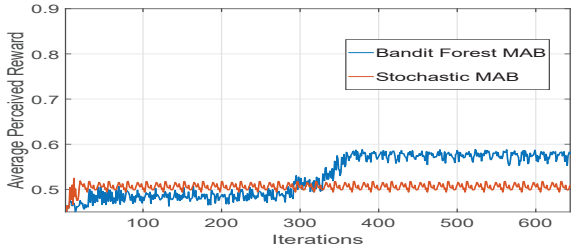
In terms of speed of convergence, the stochastic MAB SE algorithm seems to converge faster than the contextual BF. This difference is explained by the fact that the stochastic MAB, unlike the BF algorithm, does not distinguish contexts, leading indeed to faster convergence, but to sub-optimal policies as can be seen in figures 4 and 5. Moreover, the stochastic MAB converges to a single action policy as stated previously. Such policies cannot be transferable and applied in other regions of the network, as the policy's output is invariable and not adaptable with the context. The convergence time difference



(a) $w_1 = 1$

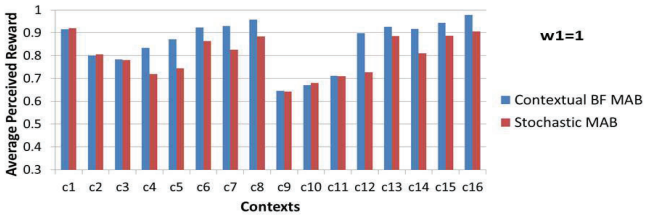


(b) $w_1 = w_2 = 0.5$

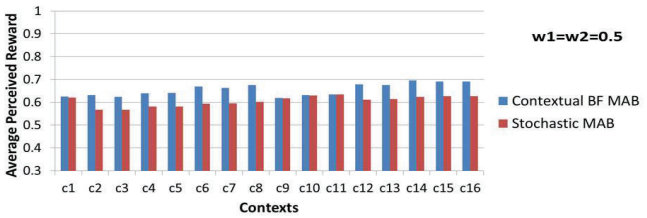


(c) $w_1 = w_2 = w_3 = 0.333$

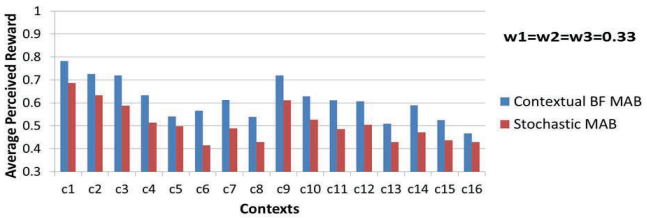
Fig. 4: Perceived Rewards Comparison



(a) $w_1 = 1$



(b) $w_1 = w_2 = 0.5$



(c) $w_1 = w_2 = w_3 = 0.333$

Fig. 5: Average Perceived Rewards per Observed Context

can be better understood by referring to [21], where the authors show that the BF's sample complexity scales exponentially with the depth of the decision trees. As a matter of fact, the MAB SE algorithm can be seen as a BF algorithm with tree depth equals to zero.

Finally, the slow convergence compared to the stochastic algorithm should not be a concern in practical cases because the AI agent can learn simultaneously from different learning clusters in the network, and constructs a common knowledge database for all of them. Having a large number of learning clusters will reduce considerably the learning time, making hence possible the deployment of such learning processes on real network.

VI. Conclusion

In this paper, we propose a context aware C-PBSM based on contextual MAB. The proposed approach consists of a centralized learning agent and a centralized knowledge database. The agent learns optimal policies simultaneously on different network sections (learning clusters), in different network locations and each having different contexts. The proposed C-PBSM is able to adapt with traffic variations in the network. The knowledge database is built collaboratively between the different learning clusters and we argue that since the learned policies are context aware, they can be transferred and applied to other network sections directly in open loop, without the need of new learning phases. We simulate a use case with different SON functions, and we implement the BF contextual MAB algorithm. Simulation results have shown that the context aware policies are globally better than a stochastic approach, when different contexts are observed in the network. We conclude that the context aware C-PBSM outperforms the C-PBSM based on stochastic MAB, by adapting its action according to the observed contexts.

Future works will focus on enhancing the context aware C-PBSM with advanced monitoring of traffic, automated detection of traffic changes including slow trend changes and sudden context shifts. Moreover, so far we consider that the C-PBSM learns through RL from scratch. It builds its knowledge only by interacting with the network. However, operators possess historic databases about network configurations and their impact. These can be of use so that the C-PBSM does not learn from scratch, but instead exploits the operator's knowledge in order to accelerate its learning process.

References

[1] Cisco, Cisco Visual Networking Index. "Global Mobile Data Traffic Forecast Update, 2018-2023." *white paper*, 2020.

[2] S.S.Mwanje, J. Ali-Tolppa and I. Malanchini, "System Aspects for Cognitive Autonomous Networks. Towards Cognitive Autonomous Networks: Network Management Automation for 5G and Beyond," 2020
[3] 3GPP TS 32.521, "UMTS; LTE; Telecommunication management; SON; NRM; IRP; Requirements," 2013.
[4] S. Hämäläinen et al., "LTE self-organising networks (SON): network management automation for operational efficiency," *John Wiley & Sons*, 2012.
[5] SEMAFOUR project, http://fp7-semafour.eu/, 2015.
[6] C. Frenzel, S. Lohmüller and L.C. Schmelz, "Dynamic, context-specific SON management driven by operator objectives," *IEEE Network Operations and Management Symposium (NOMS)*, 2014.
[7] S. Lohmüller, et al., "SON function performance prediction in a cognitive SON management system," *IEEE Wireless Communications and Networking Conference Workshops (WCNCW),* pp. 13-18, 2018.
[8] T. Daher, S. Jemaa and L. Decreusefond, "Cognitive Management of Self-Organized Radio Networks Based on Multi Armed Bandit," *IEEE Personal Indoor and Mobile Radio Communications (PIMRC)*, 2017.
[9] T. Daher, S. Jemaa and L. Decreusefond, "Linear UCB for Online SON Management," *IEEE Vehicular Technology Conference (VTC Spring)*, 2018.
[10] L.P. Kaelbling, M.L. Littman and A.W. Moore "Reinforcement learning: A survey," *Journal of artificial intelligence research,* vol. 4, pp. 237-285, 1996.
[11] S.S. Mwanje and A. Mitschele-Thiele, "A q-learning strategy for lte mobility load balancing," *IEEE Personal Indoor and Mobile Radio Communications (PIMRC)*, 2013.
[12] M. Dirani and A. Altman, "A cooperative reinforcement learning approach for inter-cell interference coordination in OFDMA cellular networks," *IEEE Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2010.
[13] P. Coucheney, K. Khawam and J. Cohen, "Multi-armed bandit for distributed inter-cell interference coordination," *IEEE International Conference on Communications (ICC)*, 2015.
[14] M. Qin et al., "Machine learning aided context-aware self-healing management for ultra dense networks with QoS provisions," *IEEE Transactions on Vehicular Technology*, 67(12), pp.12339-12351, 2018.
[15] M. Lelarge, A. Proutiere and M.S. Talebi, "Spectrum bandit optimization," *IEEE Information Theory Workshop (ITW)*, 2013.
[16] A. Feki and V. Capdevielle, "Autonomous resource allocation for dense lte networks: A multi armed bandit formulation," *IEEE Personal Indoor and Mobile Radio Communications (PIMRC)*, 2011.
[17] H. Wang et al., "Understanding mobile traffic patterns of large scale cellular towers in urban environment," *ACM Conference on Internet Measurement*, 2015.
[18] P. Auer, N. Cesa-Bianchi and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, 47(2-3), pp.235-256, 2002.
[19] L. Zhou, "A survey on contextual multi-armed bandits," arXiv preprint arXiv:1508.03326, 2015.
[20] L. Li, W. Chu, J. Langford and R.E. Schapire, "A contextual-bandit approach to personalized news article recommendation," *In Proceedings of the 19th international conference on World wide web*, pp. 661-670, ACM, 2010.
[21] R. Féraud, R. Allesiardo, T. Urvoy, and F. Clérot, "Random forest for the contextual bandit problem," *In Artificial Intelligence and Statistics*, pp. 93-101, 2016.
[22] W. Chu, L. Li, L. Reyzin and R. Schapire, "Contextual bandits with linear payoff functions," *In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208-214, 2011.
[23] A. Slivkins, "Contextual bandits with similarity information," *Journal of Machine Learning Research*, 15(1), pp.2533-2568, 2014.
[24] T. Lu, D.Pál and M. Pál, "Contextual multi-armed bandits," *In Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, pp. 485-492, 2010.
[25] E. Even-Dar, S. Mannor and Y. Mansour, "PAC bounds for multi-armed bandit and Markov decision processes," *In International Conference on Computational Learning Theory*, pp. 255-270, Springer, 2002.
[26] L. Breiman, "Classification and regression trees," *Routledge*, 2017.
[27] P. Domingos and G. Hulten, "Mining high-speed data streams," *In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 71-80, 2000.
[28] L. Brieman, Random forests. Machine learning, 45(1), pp.5-32, 2001.
[29] E. Even-Dar, S. Mannor and Y. Mansour, "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems," *Journal of Machine Learning Research*, pp.1079-1105, 2006.