Learning to Dynamically Allocate Radio Resources in Mobile 6G in-X Subnetworks

Ramoni Adeogun[(1)], Gilberto Berardinelli[(1)] Preben Mogensen[(1,2)]
[(1)] *Department of Electronic Systems, Aalborg University, Denmark*
[(2)] *Nokia Bell Labs, Aalborg, Denmark*
E-mail:{ra, gb, pm}@es.aau.dk

***Abstract*—This paper investigates efficient deep learning based methods for interference mitigation in independent wireless subnetworks via dynamic allocation of radio resources. Resource allocation is cast as a mapping from interference power measurements at each subnetwork to a class of shared frequency channels. A deep neural network (DNN) is then trained to approximate this mapping using data obtained via application of centralized graph coloring (CGC). The trained network is then deployed at each subnetwork for distributed channel selection. Simulation results in an environment with mobile subnetworks have shown that relatively small-sized DNNs can be trained offline to perform distributed channel allocation. The results also show that depending on the choice of initialization, a DNN for distributed channel selection can achieve similar performance as CGC in diverse environments with only aggregate interference power measurements as input.**

***Index Terms*—6G, resource allocation, machine learning, 5G, industrial automation, intra-vehicular communication, in-X subnetworks**

I. INTRODUCTION

Wireless systems are continuously faced with the demand to support higher reliability, lower latency, increased data rate and improved coverage. Consequently, beyond 5th Generation (5G) systems may need to support up to 10x lower latency and higher reliability than the 1 ms and 99.9999% limits in 5G. For example, industrial closed-loop control at the sensor-actuator level may require sub-milliseconds communication latencies [1] with extremely high reliability in order to preserve stability of the control loop. Similar extreme connectivity requirements may also be demanded in other evolving life-critical wireless-applications such as brake and ignition control in intra-vehicle communication, wireless heart pace-maker in intra-body networks and intra-avionics communication [2].

Although use-cases and applications that may feature such extreme requirements are still evolving, recent visions on 6th Generation (6G) networks [3]–[5] have identified independent and uncoordinated *subnetworks* (i.e., short range cells comprising of a controller acting as the access point for multiple devices) as potential solutions for supporting extreme connectivity. In [2], visions and design concepts for such 6G in-X subnetworks are presented. The term in-X for inside-everything was introduced in [6] to highlight the emerging scenarios such as in-robots, in-vehicles, in-aircrafts, and in-human bodies where these subnetworks are expected to be installed. These use cases can lead to situations with fast-moving subnetworks and hence highly dynamic interference conditions. Also, the lack of coordination may lead in some cases to high interference power translating to higher failure rates than tolerated. Thus, efficient and robust algorithms that are capable of adapting utilization of the available multi-dimensional radio resources (such as frequency bands, time slots and transmit power) to dynamic interference conditions under super-tight latency constraints are crucial for these systems. Since 6G subnetworks are expected to operate with very low power (e.g., -10dBm per channel in [6]), optimizing power usage may not yield any significant gains. We therefore focus on methods for dynamically managing the time-frequency resources.

Conventionally, resource allocation is formulated as mathematical optimization problems which can be solved online based on instantaneous measurements of selected wireless environment variables, see e.g., [7]–[9]. However, the non-convex nature of most resource allocation problems often leads to cumbersome and sometimes intractable procedure for obtaining optimal solutions. To overcome these limitations, existing research works often rely on heuristic algorithms for solving resource allocation problems which in most cases yield sub-optimal solutions. In general, existing solutions can be broadly classified into coordinated or uncoordinated algorithms. Coordinated algorithms are based on explicit inter-cell interference coordination and typically assume the existence of a communication link to a central resource manager or among different cells. On the other hand, uncoordinated methods are purely distributed and require no centralized management or exchange of information among cells. Clearly coordinated schemes are not realizable in wireless networks with independent subnetworks necessitating the need for efficient distributed algorithms for resource allocation.

The work in [10] presented three heuristic algorithms viz: ϵ-greedy selection, minimum signal to interference plus noise ratio (SINR) guaranteed and Nearest Neighbour Conflict Avoidance (NNCA) for distributed Dynamic Channel Allocation (DCA) in mobile independent subnetworks based on aggregate interference sensing measurements. The NNCA additionally requires accurate identification of the channels occupied by the nearest neighboring subnetworks. The results indicated that distributed NNCA algorithm can provide more than two-fold reduction in required bandwidth relative to static allocations for achieving a low target failure rate. Nonetheless, this comes at the expense of accurate identification of the neighbour subnetworks; obtaining such identity information will require additional system overhead and complex receiver processing, making this algorithm unattractive for practical implementation.

Motivated by recent advances in machine learning and its applications to different wireless communication problems, see e.g., [11]–[14], we investigate supervised learning methods for efficient DCA with limited sensing information in this paper. The goal is to develop a fully distributed learning based algorithm with similar or better performance than existing distributed algorithms but using only measurements of the aggregate interference power at each subnetwork for channel selection decisions and hence, eliminating the need for the costly subnetwork identification procedure.

We propose a novel Deep Neural Network (DNN) based distributed DCA method for mobile independent subnetworks. The proposed method involve offline training data generation using a centralized graph coloring (CGC) algorithm, DNN architecture design and training and a distributed execution for interference power - channel selection mapping. It should be noted that a centralized algorithm is not realizable for the considered scenarios with independent subnetworks thereby making usage of CGC for channel allocation in such scenarios impossible. We however, assume that such centralized scheme provide a reasonable benchmark for distributed DCA performance and hence a suitable choice for offline training data generation. The main contributions of this paper include:

- We design a DNN that is capable of learning to map aggregate interference power measurements at each mobile subnetwork to channel selection in a distributed version based on simulated training examples obtained via application of CGC.
- We show via simulations that a DNN can be trained to perfom channel allocations in a distributed version based on aggregate interference power measurements generated using CGC with up to 80% accuracy and a mean absolute power difference of about 0.7 dB.
- The DNN based algorithm is applied to a network of 6G in-X subnetworks and its performance compared to that of existing methods under different propagation conditions and initialization procedures.

We remark that although the DCA method is presented in the context of 6G in-X subnetworks, it can be applied to other wireless systems with uncoordinated deployments.

II. System Model

We consider a network comprising of a set of N independent and asynchronous mobile subnetworks each having M sensor-actuator pairs and a single controller as illustrated Fig 1. Each subnetwork (i.e, controller and devices) moves at a specified speed in a random direction. This can for example represent subnetworks installed in mobile robots or inside moving vehicles [2]. The controller periodically receive measurements from the sensors and then generate appropriate command to the actuators. We will refer to the sensor-to-controller and controller-to-actuator transmission as uplink (UL) and downlink (DL), respectively. A combination of UL and its associated DL is referred to as a *loop*. Each subnetwork is expected to guarantee extreme connectivity with outage probability below 10^{-6} for all communication loops at every spatio-temporal instance.

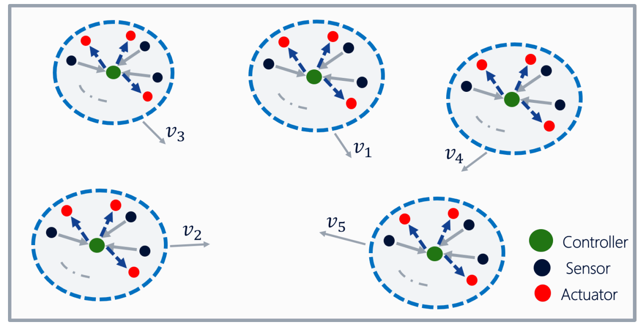


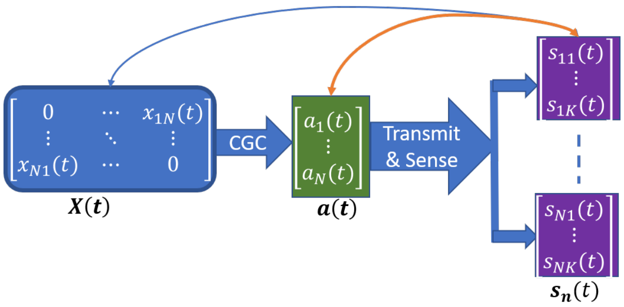
Fig. 1. Illustration of a deployment with 5 mobile subnetworks.

We assume that each packet is mapped into a fixed payload and that transmissions are performed periodically using the Medium Access Control (MAC) design with a symmetric Time Division Duplexing (TDD) frame structure proposed in [6]. In the frame structure, the total bandwidth, B is partitioned into N_{ch} equal channels. Each UL and DL subframe is divided into N_{tu} time units (TUs). Each TU corresponds to the continuous transmission time by a device over a given channel. We further consider blind repetitions of each packet by hopping over multiple frequency channels in order to improve communication reliability and harvest frequency diversity gain. It is also assumed that transmissions within a subnetwork are orthogonal and hence there exist no intra-subnetwork interference.

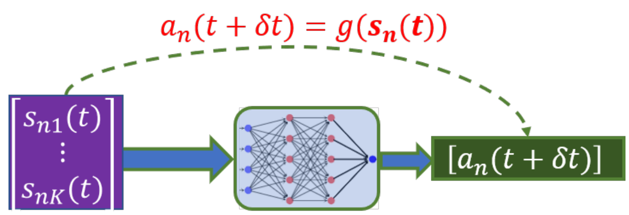
Given the lack of synchronization among subnetworks, there is a high probability of mutual interference between UL and DL transmissions besides same-link interference as in time-aligned networks. Thus, the SINR on each UL (DL) is calculated as the ratio of the desired power to the sum of the the noise power and the aggregate UL-UL (DL-UL) and DL-DL (UL-DL) interference power. We are interested in dynamic allocation of frequency channels such that an outage probability below a specified target, $P_{\mathrm{out,T}}$ is achieved for all loops with cycle time below 0.1 ms. Similar to [10], we consider a block fading channel model with capacity achieving codes and chase combining of the multiple repetitions. Assuming reception over N_{rx} uncorrelated antennas, the outage probability after all N_{rep} repetitions can then be written as [6]

$$P_{\mathrm{out}} = \prod_{u=1}^{N_{\mathrm{rep}}} \Pr\left[\frac{1}{L}\sum_{\ell=1}^{L}\log_2\left(1+\sum_{z=1}^{N_{\mathrm{rx}}}\gamma_{\ell,z}\sum_{p=1}^{u}\mathrm{sinr}_p\right) < R\right]. \quad (1)$$

where L is the number of fading blocks, R is the transmission rate, sinr_p is the average SINR on the pth channel and $\gamma_{\ell,z} = |h_{\ell,z}|^2$ is the small scale power for the zth receive antenna on the ℓth fading block with $h_{\ell,z}$ denoting the small scale fading gain.



(a)



(b)

Fig. 2. The data generation procedure (a) and an illustration (b) of the proposed DNN mapping of sensing measurements of aggregate interference power at the nth subnetwork to channel selection.

The task, therefore, is to ensure that $P_{\text{out}} \leq P_{\text{out,T}}$. To support repetitions over different frequency channels, each subnetwork needs to select N_{rep} out of the N_{ch} available channels for transmission resulting in high channel decision signalling overhead. Moreover, such combinatorial problem may easily become intractable. To eliminate this problem, we instead partition the N_{ch} channels into $K = N_{\text{ch}}/N_{\text{rep}}$ groups each with N_{rep} channels. Each subnetwork can then operate over a single channel group at any given time and only switches to another group when the radio conditions are estimated not to be sufficiently good to guarantee the outage probability target. We refer to our previous works [6], [10] for further details on the system model.

III. Dynamic Channel Allocation

In this section, we introduce the CGC algorithm for generating training data and present the proposed DNN based distributed DCA procedure.

A. Centralized Graph Coloring Algorithm

The CGC method utilizes a graph coloring algorithm for color assignment such that nearest neighbours do not share a common channel group. Recall that as a result of the uncoordinated deployments for 6G subnetworks, a centralized algorithm is not realizable in practice. The CGC involves the following steps:

1) Interference Graph Creation: We assume that at each time instant, t, measurements of the pair-wise interference between subnetworks can be collected into an $N \times N$ matrix, $\mathbf{X}(t)$. Using $\mathbf{X}(t)$, a conflict graph, G_t with subnetworks as vertices and edges defined by connecting each vertex, v to $K-1$ other vertices generating the strongest interference to v is created.

Algorithm 1 Improper Graph Coloring Procedure

Input: Interference matrix, $\mathbf{X}(t)$, number of channel groups, K
Create conflict graph, G_t
Apply greedy coloring, $C \leftarrow \text{GreedyColor}(G_t, K)$
while $\max(C) > K$ **do**
    Remove edge with lowest interference power in G_t
    Re-apply greedy coloring, $C \leftarrow \text{GreedyColor}(G_t, K)$
end while
Output: Assigned colors , C

2) Vertex coloring: Coloring of G_t is performed at every update instant using a greedy algorithm [15]. The goal is to obtain a centralized assignment which guarantees that the number of colors is upper bounded by the number of available channels. This is therefore an improper coloring problem which is known to be NP-complete [16]. Our initial experiments also indicated that instances where K-coloring is not achievable may arise albeit with very small probability. To guarantee K-coloring of the conflict graph, we propose a sparsity inducing procedure involving successive removal of edges with minimum interference power until K-coloring is achieved. The vertex coloring procedure is summarized in Algorithm 1. The assigned colors, C by the algorithm correspond to the channel group assignment for all subnetworks, i.e., $\mathbf{a}(t) = [a_1(t), a_2(t), \cdots, a_N(t)]^T$, where $(\cdot)^T$ denotes the transpose operator.

B. DNN for Distributed Dynamic Channel Allocation

Relying on the observations in [12]–[14], we assume that a functional relationship $g(\cdot)$ exists between the cumulative interference power measurements at each subnetwork and the assigned color by the CGC algorithm. Thus, for the nth subnetwork, channel selection at transmission instant t can be expressed as $a_n(t+\delta t) = g(\mathbf{s}_n(t))$, where δt denotes the time interval between channel selection and actual usage. Based on the universal approximation theorem [17], we conjecture that a DNN can be trained to approximate this relationship. We adopt a classification approach in which the available channel groups are the classes. The input to the DNN is the cumulative interference power vector $\mathbf{s}_n(t) = [s_{n1}(t), s_{n2}(t), \cdots, s_{nK}(t)]^T$, with n denoting the index of a generic subnetwork, and its output is the assigned channel group for the next transmission, $a_n(t+\delta t)$ (where δt is the time interval between channel selection decision and the next transmission) as shown in Fig. 2(b). Observe that, though the centralized training required information of the conflict graph $\mathbf{X}(t)$ of the entire network, the mapping in the execution phase is performed individually at each subnetwork as it requires only information of its own measured aggregated interference power. The channel selection for subnetwork n at time instant t can be expressed using a DNN architecture with L hidden layers as

$$\hat{a}_n(t+\delta t) = \arg\max_{a}(\omega_s(\mathbf{W}_o\mathbf{h}_L + \mathbf{b}_o)), \tag{2}$$

where $\mathbf{W}_o$ and $\mathbf{b}_o$ are the weights and biases of the output layer, respectively. ω_s and the output layer's softmax activation function and $\mathbf{h}_L$ denote the output of the Lth hidden layer. The output of the lth hidden layer can be written as

$$\mathbf{h}_l = \omega_h(\mathbf{W}_l\mathbf{h}_{l-1} + \mathbf{b}_l); \quad l = 1, 2, \cdots, L, \tag{3}$$

with $\mathbf{h}_0 = \mathbf{s}_n(t)$. $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weights and biases of the lth hidden layer. Due to its non-vanishing gradient property, the Rectified Linear Unit (ReLU) is used at the hidden layers, i.e., $\omega_h(x) = \max(x, 0)$; where x denotes the input to a neuron's activation function. As shown in (2) and (3), a major part of the DNN based channel allocation is the estimation of $\{\mathbf{W}_o, \mathbf{b}_o, \{\mathbf{W}_l, \mathbf{b}_l\}_{l=1}^{L}\}$ via training. The data generation, network training and deployment procedures are described in the sequel.

C. Data Generation

The proposed data generation procedure at time instant, t, is illustrated in Fig. 2(a). At each time instant, the coloring procedure in Algorithm 1 is applied on the conflict graph, G_t obtained from the interference power matrix, $\mathbf{X}(t)$ to obtain channel assignments for all subnetworks, $\{a_n(t)\}_{n=1}^{N}$. The aggregate interference power on all channels at each subnetwork, $\{\mathbf{s}_n(t)\}_{n=1}^{N}$ is then calculated. This process is repeated over a simulation time with M_t transmission instants. The aggregate interference power obtained at all subnetworks and the corresponding channel assignments are then collected into a training data matrix and label matrices, $\mathbf{S} \in \mathbb{R}^{K \times M_t N}$ and $\mathbf{A} \in \mathbb{R}^{1 \times M_t N}$, respectively. These matrices contain $M_t N$ training examples. A training example is a pair of interference power vector, $\mathbf{s}(t) \in \mathbf{R}^{K \times 1}$ at each subnetwork (corresponding to a column of $\mathbf{S}$ and the assigned channel group, a by the CGC (corresponding to an element in $\mathbf{A}$).

D. DNN Parameters Optimization

The DNN parameters are optimized using $M_t N$ training examples in $\{\mathbf{S}, \mathbf{A}\}$. We adopt a mini-batch gradient descent procedure in which the training data is divided into batches. The DNN parameters are then optimized by minimizing the cross-entropy loss between the predicted channel group by DNN, $\hat{a}_n(t)$ and the CGC's assigned group, $a_n(t)$ using a suitable gradient descent algorithm. The DNN's prediction is evaluated using classification accuracy and mean absolute power difference (MAPD) as metrics. The MAPD is introduced to access the communication theoretic performance and is defined as the absolute difference between the interference power on the channel assigned by CGC and that predicted by the DNN averaged over the data set.

E. Distributed Channel Allocation

Once trained, the DNN is deployed at each subnetwork for distributed channel selection. The controller at each subnetwork continuously estimates the SINR on its occupied channel group and performs sensing to acquire measurements of the aggregate interference power level on all groups. If a new channel group is selected, the controller signals its decision to all devices to enable transmission on the new channel group.

TABLE I
SIMULATION PARAMETERS.

Parameter	Value
Deployment and system parameters	
Deployment area [m^2]	30 $\times$ 30
Number of controllers/subnetworks, N	16
Number of devices per subnetwork, M	18
Cell radius [m]	2.5
Velocity, v [m/s]	2.0
Minimum inter-subnetwork distance [m]	1.5
Number of channels, N_{ch}	12
Number of groups, N_{gr}	6
Number of receive antenna	2
Propagation and radio parameters	
Pathloss exponent, ϵ	1.8/2.2
Shadowing standard deviation, σ_s [dB]	3/5
De-correlation distance, d_c [m]	4
Lowest frequency [GHz]	6
Transmit power per channel [dBm]	-10
Noise figure [dB]	10
Subcarrier spacing [kHz]	480
Payload size [bytes]	50
Per channel bandwidth [MHz]	40 - 320
DNN and simulation settings	
Number of hidden layers	2
Number of neurons per layer	30
Optimizer	Adam
Learning rate	0.01
Batch size	32
Training duration [s]	600
Simulation time [s]	2000
Snapshot duration [s]	20
Measurement update interval [ms]	5

We remark here that the DNN based scheme is particularly feasible if the environment where the subnetworks are to operate can be simulated with a reasonable degree of accuracy, in order to generate sufficiently accurate training data. This can be the case, for example, of an indoor factory scenario where the propagation environnment can be studied and modelled beforehand.

IV. PERFORMANCE EVALUATION

A. Simulation settings

We now evaluate the performance of the DNN based scheme and compare with the algorithms in [10] using a snapshot based procedure. We consider a network with 16 subnetworks each with 18 sensor-actuator pairs in a 30 m $\times$ 30 m rectangular deployment area. We consider the symmetric TDD frame structure in [6] with 90 μs total duration and 12 frequency channels over a total bandwidth , B. Thus, each loop, i.e., ensemble of UL (sensor-to-controller) and DL (controller-to-actuator) transmission is completed within the 90 μs duration.

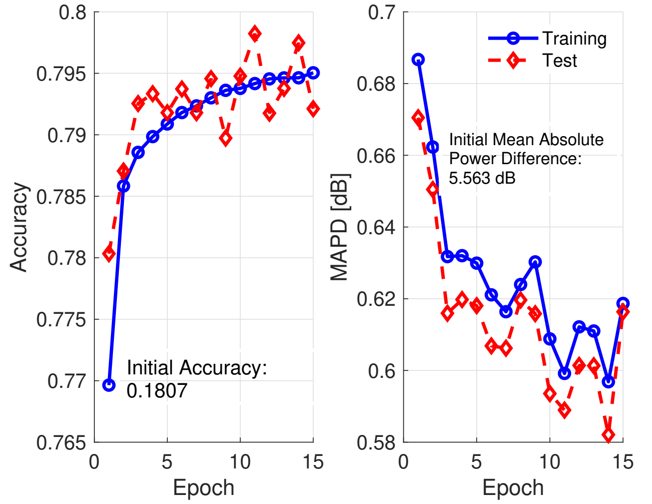


Fig. 3. Channel selection accuracy and MAPD between DNN prediction and the assigned channels by the CGC baseline.

Each subframe is partitioned into 18 TUs each with 2.5 μs duration and the bandwidth is divided into 6 groups each with two channels translating to a maximum of 18 devices with 2 repetitions per device. For each transmission, we mapped a fixed payload onto a single OFDM symbol with 480 kHz subcarrier spacing. Other simulation parameters are shown in Table I.

We used the reference distance path-loss model with parameters set based on a recent channel measurements in typical industrial environments [18]. Temporal and spatial correlated shadow fading are generated using the Gaussian random fields based model in [19]. Small scale fading is assumed to be Rayleigh distributed. A random mobility which begins with uniform distribution of the subnetworks within a rectangular deployment area at each snapshot is used in the simulations. Each subnetwork then moves with a fixed speed, v = 2 m/s, in a random direction. The direction is changed when a subnetwork reaches a boundary or it's within $\leq$ 1.5 m distance from any other subnetwork. The latter eliminate unrealistic collision of subnetworks in the simulation. We consider 8 configurations with evenly spaced bandwidth per channel between 40 MHz and 320 MHz.

B. Training data generation and performance

To generate training data, we set pathloss exponent, $\epsilon = 2.2$, shadow fading standard deviation, $\sigma_{\rm s} = 3$ dB and decorrelation distance, $d_c = 4$ m. We then perform simulation for a total duration of 600 s with the CGC. The interference graph and color assignment are updated every 5 ms. At each update instance, the aggregate interference power on all channel at each subnetwork is calculated translating to a total of 1920000 interference power - channel pairs over the entire duration. This data is then used to train a DNN classification network using Adam [17] with learning rate and batch size of 0.01 and 64, respectively, for optimizing the network weights. The DNN architecture and training parameters were selected via a procedure which involves comparison of learning curves with different network architectures and a range of values of learning rate and batch size. Details of this initial network architecture and parameter selection are not shown here due to space constraints. The analysis indicated that there is no noticeable improvement from having more than two hidden layers. We therefore, use a 2-hidden layers architecture for the performance evaluations.

In Fig. 3, we show the accuracy and the corresponding MAPD on both the training and test data sets. The figure shows accuracy and MAPD of approximately 80% and 0.6 dB. The small power difference indicates that the selected channels by DNN are always good even in the 20% instants where the prediction is not same as the CGC assignment.

C. Performance results

We now apply the trained DNN for distributed channel allocation and compare obtained communication performance with the CGC as well as the heuristic algorithms in [10] viz:

- Random: select a new channel randomly,
- Greedy: select the channel with the least aggregate interference power,
- Nearest Neighbour Conflict Avoidance (NNCA): select a channel that is not occupied by $K-1$ nearest subnetworks.

As mentioned in the introduction, while the random and greedy algorithms are based solely on aggregate interference power measurements at each controller, the NNCA additionally require identification of the channels occupied by nearest neighbours.

Performance is evaluated using the probability of loop failure (PLF) defined as the ratio of the number of loops with outage probability on both UL and DL greater than or equal to $P_{\rm out,T} = 10^{-6}$ to the total number over the entire simulation time, with $P_{\rm out,T}$ calculated as in (1). The DCA algorithms require additional resources for accommodating switching delay and signalling switching decisions. In order to achieve reasonable resource utilization efficiency, it is therefore necessary to minimize this overhead. As a measure of the resource overhead for enabling DCA using the algorithms, we compare the averaged time between channel switching. A low time between channel switching corresponds to high switching frequency and hence, high overhead. Thus, the target is to obtain an algorithm with good performance and high time between channel switching.

At the beginning of the simulation, the distributed schemes are initialized with either randomly selected actions or via application of CGC to the inter-controller distance matrix. The latter is only possible in controlled environment, where an interference graph can be built for initializing the channel group selection. During the simulation, SINR and interference power measurements at each subnetwork are updated at an interval of 5 ms. Each subnetwork then perform channel switching using the algorithms if the minimum SINR on all its transmissions is below a specified decision threshold which is

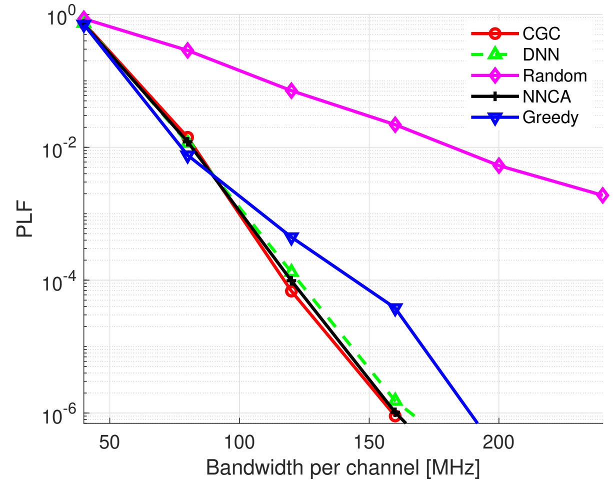


Fig. 4. Probability of loop failure versus bandwidth/channel with initialization using CGC assignment.

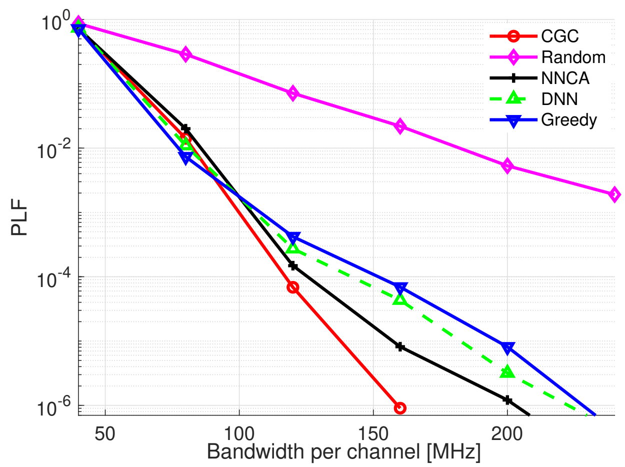


Fig. 5. Probability of loop failure versus bandwidth/channel with random initialization.

set for each bandwidth configuration as the minimum SINR at which $P_{\text{out}} \leq 10^{-6}$ plus a margin of 3 dB. The thresholds are calculated beforehand based on (1), for the different bandwidth configurations and assuming the parameters in Table I. We introduce a random switching delay in order to minimize potential *ping-pong* effects resulting from multiple subnetworks switching to the same channel group simultaneously. This delay is generated for each snapshot as a random integer factor (between 1 and 8) of the update interval.

Fig. 4 shows the PLF versus per channel bandwidth configurations for the DNN and other algorithms with all distributed schemes with CGC initialization. The figure shows significant reduction in PLF with DCA algorithms relative to the the fully random scheme translating to reduction in the bandwidth required for supporting the below 10^{-6} outage probability target up to a desired PLF. Of all distributed algorithms, the DNN based method shows the best performance - similar to CGC and NNCA. This is expected considering the low power difference between the DNN predictions and colors assigned by CGC in Fig 3. It is worth to recall that, for NNCA, each subnetwork additionally requires identification of channels occupied by nearest neighbours leading to high overhead and additional signal processing compared to the DNN approach which is based solely on aggregate interference measurements. Moreover, the DNN approach offers the flexibility for online tuning and hence, adaptation to varying propagation conditions.

The results presented so far have shown that the DNN based method for distributed DCA provides similar performance to the CGC baseline but with only limited sensing information - the aggregate interference power vector at each subnetwork. The analysis relied on the assumption that it is possible to synthetically generate sufficiently large amount of data which is representative of the environment where the subnetworks are expected to operate and that CGC initialization is possible.

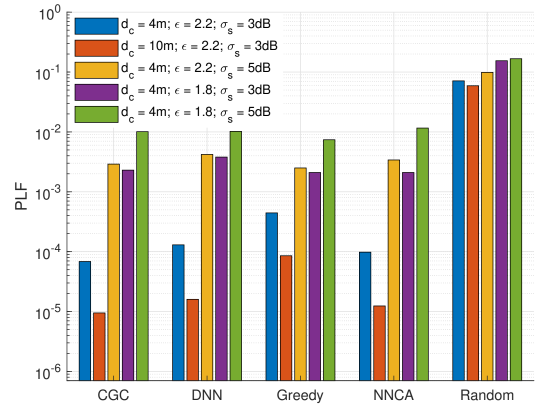


Fig. 6. PLF performance in environment with varying propagation parameters and random initialization.

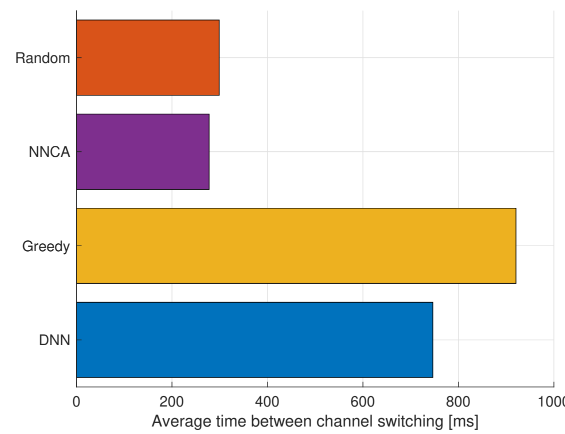


Fig. 7. Averaged time between channel switching for the distributed algorithms at 80 MHz per channel bandwidth and with random initialization.

However, as a result of the dynamic nature of wireless propagation environments, it is nearly impossible to collect such representative dataset. Thus, the obtained performance in real deployment scenarios may be degraded due to discrepancies between the simulated environment used for training data generation and real propagation scenarios. It is therefore useful to evaluate sensitivity of the DNN based method to changes in propagation parameters as well as the choice of initialization.

In Fig. 5, we compare the PLF performance for all algorithms with random initialization. Compared to the Fig. 4, the figure shows that the choice of initialization impact performance of the distributed schemes but only at the low percentile (below 2×10^{-3}) of the PLF. In the DNN, a potential approach for circumventing this performance degradation is online hyper-parameter tuning. This is however left for future work.

In Fig. 6, we illustrate the robustness of the DNN based algorithm to varying wireless environment conditions by applying the DNN, NNCA and CGC to scenarios with path-loss exponent, shadow fading standard deviation and/or the correlation distance different from the training settings. The figure indicates that increasing the correlation distance results in improved performance relative to Fig. 4. In contrast, increased shadowing and decreased path-loss exponent result in degradation of the PLF performance. The figure also shows that all algorithms are equally affected by changes in the environment. This is an indication that the DNN generalizes well to propagation conditions which are different than the ones experienced in the training phase.

Finally, we compare the time between channel switching averaged over all subnetworks for the different schemes in Fig. 7 with random initialization. The figure shows that at 80 MHz per channel bandwidth, the greedy and DNN schemes have the highest inter-channel switching time of about 920 ms and 750 ms, respectively. The NNCA has the lowest time between switching translating to highest overhead. This relatively high overhead coupled with the additional subnetwork identification requirement makes the NNCA less-attractive for practical implementation.

V. Conclusion

The application of deep learning for distributed dynamic channel allocation in 6G in-X subnetworks is investigated in this paper. We propose a method involving training of a DNN classification network to perform distributed allocation using data from centralized graph coloring algorithm. Performance evaluation results show that a suitably trained DNN can achieve similar performance as the centralized baseline even in environments with propagation conditions different from the ones used for generating the training data while requiring only local measurements at each subnetwork of the aggregated interference power over the channels. It is further observed that the choice of initialization is crucial in order to achieve the extremely low probability of loop failure for the envisioned use-cases for the 6G in-X subnetworks.

Acknowledgment

This work is supported by the Danish Council for Independent Research, grant no. DFF 9041- 00146B.

References

[1] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 88–145, 2019.

[2] G. Berardinelli, P. Mogensen, and R. O. Adeogun, "6G subnetworks for Life-Critical Communication," in *2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5.

[3] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The Roadmap to 6G: AI Empowered Wireless Networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[4] H. Viswanathan and P. E. Mogensen, "Communications in the 6G Era," *IEEE Access*, vol. 8, pp. 57 063–57 074, 2020.

[5] V. Ziegler, H. Viswanathan, H. Flinck, M. Hoffmann, V. Räisänen, and K. Hätönen, "6G architecture to connect the worlds," *IEEE Access*, vol. 8, pp. 173 508–173 520, 2020.

[6] R. Adeogun, G. Berardinelli, P. E. Mogensen, I. Rodriguez, and M. Razzaghpour, "Towards 6G in-X Subnetworks With Sub-Millisecond Communication Cycles and Extreme Reliability," *IEEE Access*, vol. 8, pp. 110 172–110 188, 2020.

[7] R. O. Adeogun, "A novel game theoretic method for efficient downlink resource allocation in dual band 5G heterogeneous network," *Wireless Personal Communications*, vol. 101, no. 1, pp. 119–141, Jul 2018.

[8] Jianping Jiang, Ten-Hwang Lai, and N. Soundarajan, "On distributed dynamic channel allocation in mobile cellular networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 13, no. 10, pp. 1024–1037, 2002.

[9] E. Kudoh and F. Adachi, "Distributed dynamic channel assignment for a multi-hop virtual cellular system," in *IEEE VTC Spring*, vol. 4, 2004, pp. 2286–2290 Vol.4.

[10] R. Adeogun, G. Berardinelli, I. Rodriguez, and P. E. Mogensen, "Distributed Dynamic Channel Allocation in 6G in-X Subnetworks for Industrial Automation," in *IEEE Globecom Workshops*, 2020.

[11] D. Kwon, J. Jeon, S. Park, J. Kim, and S. Cho, "Multi-Agent DDPG-based Deep Learning for Smart Ocean Federated Learning IoT Networks," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[12] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to Optimize: Training Deep Neural Networks for Interference Management," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[13] C. Sun and C. Yang, "Learning to Optimize with Unsupervised Learning: Training Deep Neural Networks for URLLC," in *IEEE PIMRC*, 2019, pp. 1–7.

[14] F. Zhou, X. Zhang, R. Q. Hu, A. Papathanassiou, and W. Meng, "Resource allocation based on deep neural networks for cognitive radio networks," in *IEEE/CIC International Conference on Communications in China (ICCC)*, 2018, pp. 40–45.

[15] L. Ouerfelli and H. Bouziri, "Greedy algorithms for dynamic graph coloring," in *International Conference on Communications, Computing and Control Applications (CCCA)*, 2011, pp. 1–5.

[16] J. A. Torkestani and M. R. Meybodi, "Graph coloring problem based on learning automata," in *2009 International Conference on Information Management and Engineering*, 2009, pp. 718–722.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[18] M. Razzaghpour, R. Adeogun, G. Berardinelli, M. Ramsmus, P. Troels, M. Preben, and S. Troels, "Short-Range UWB Wireless Channel Measurement in Industrial Environments," in *WIMOB*, 2019.

[19] P. Agrawal and N. Patwari, "Correlated link shadow fading in multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 8, pp. 4024–4036, 2009.