Traffic Scheduling Optimization in Cognitive Radio based Smart Grid Network using Mini-batch Gradient Descent Method

Muhammad Waqas Khan[1a], Muhammad Zeeshan[2a], Muhammad Usman[3b]
[a]College of Electrical and Mechanical Engineering (CEME)
National University of Sciences and Technology (NUST)
Islamabad, 44000, Pakistan
[b]Center for Advanced Research in Engineering (CARE)
Islamabad, 44000, Pakistan
Email: [1]muhammad.waqas@ceme.nust.edu.pk, [2]ranazeeshan@ceme.nust.edu.pk, [3]muhammad.usman@carepvtltd.com

***Abstract*—Smart Grid (SG) network comprises of wide variety of applications with diverse Quality of Service (QoS) requirements. Each SG application has its own requirements of latency, throughput and reliability. Cognitive Radio (CR), which offers better spectrum utilization through opportunistic spectrum access and spectrum sharing, is considered as a promising wireless technology for SG networks. Traffic scheduling and optimization is one of major challenges in CR-based SG communication network. In this paper, we propose a novel usage of mini-batch gradient descent method for the optimization of QoS-based differential scheduling in CR-based SG network. A 2-class based priority scheduling model with emergency and interrupt handling capability is used. It is shown through simulation results that mini-batch gradient descent method achieves better results as compared to gradient descent method in terms of the fast convergence and minimization of the overall cost function.**

***Index Terms*—Smart Grids, Cognitive Radio, Traffic scheduling**

I. Introduction

The traditional power grid offers unidirectional power delivery and information flow. The information monitoring is handled in the distribution grids and the power flow is only from utility centers to customer premises [1]. Smart Grid (SG) brings evolution in the conventional power grid by mitigating its drawbacks and introducing multi-directional communication within complete power system network. SG is more robust, resilient and reliable as compared to traditional power grid [2]. To achieve the true functionality of smart grid, new communication technologies need to be integrated.

Traffic scheduling and optimization is one of the major challenges on cognitive radio based smart grid. An algorithm for data priority scheduling is proposed in [3] for SG based Wireless Sensor Networks (WSNs). Firstly, the ease provided by WSNs in SG network is highlighted. Secondly, authors propose a data transmission strategy for data traffic scheduling to ensure minimum delay for critical information. Another algorithm for traffic scheduling CR-based SG network is given in [4]. This algorithm utilizes the re-routing methodology for transmissions of critical data through nearby cells. A QoS-based differential scheduling algorithm based on backpropagation neural network for CR-based SG network is proposed in [5]. A critic network based on gradient descent method is used for the optimization.

The CR technology allows efficient spectrum utilization through opportunistic spectrum access and spectrum sharing. It is based on two types of users, i.e. Primary Users (PUs), which are high priority licensed users, and Secondary Users (SUs), which opportunistically use spectrum white spaces (the unused spectrum bands). Through the use of CR technology, the SG users can efficiently utilize the complete available spectrum and thus improve spectral utilization to support explosive data traffic inherent in SG networks.

In SG networks, each application has its own QoS requirements in terms of latency, reliability and throughput. Some applications (e.g. Advanced Metering Information etc) do not require real-time information exchange, so their QoS requirement is low. On the other side, detection and reporting faults in the form of alarms belong to emergency data that need to be communicated in real-time. Due to this differential QoS requirement, an efficient traffic scheduling and optimization scheme is required which can fulfill the QoS requirements of all the SG applications. In this paper, we propose a novel usage of mini-batch gradient descent method for the optimization of QoS-based differential scheduling in CR-based SG network. A 2-class based priority scheduling model with emergency and interrupt handling capability is used. It is shown through simulation results that mini-batch gradient descent method achieves better results as compared to gradient descent method in terms of the fast convergence and minimization of the overall cost function.

II. CR-based Smart Grid Network

The complete SG network comprises of three layers; Home Area Network (HAN), Neighborhood Area Network (NAN), and Wide Area Network (WAN). Furthermore, there are six major SG applications which have versatile QoS requirements. These three layers and six applications make SG network

2019 14th Iberian Conference on Information Systems and Technologies (CISTI)
19 – 22 June 2019, Coimbra, Portugal
ISBN: 978-989-98434-9-3

highly heterogeneous [6]. Fixed spectrum allocation is usually adopted in wireless networks which can result in inefficient spectrum utilization. Cognitive radio (CR) which works on the principle of opportunistic spectrum access is a suitable wireless technology to overcome these spectrum inefficiencies [7], [8] and other networking and communication issues of SG network[9]. There are six SG applications, as per US department of energy, which are mentioned below [10];

1) Advanced Metering Infrastructure (AMI)
2) Demand Response Management (DRM)
3) Wide-Area Situational Awareness (WASA)
4) Distributed Energy Resources (DER) and Storage
5)]Electric Transportation
6) Distribution Grid Management (DGM)

III. QoS Differential Transmission Scheduling

In this paper, we have used 2-class based priority transmission scheduling model proposed in [5]. The SG users, who provide critical messages related to fault reporting, protection, control etc., belong to the higher priority class (c1). Those, who carry less critical data e.g. meter reading or similar data, belong to the lower priority class (c2). In this way, the data transmission from class c1 SGUs has higher priority as compared to class c2 SGUs. However, in case class c2 SGUs have some emergency data, they get higher priority as compared to non-emergency class 1 SGUs. If both classes have emergency data to transmit, higher priority is given to class c1, as expected. Another scenario which is incorporated in this 2-class model is the high weightage given to interrupted data transmissions. It means that within the same class, higher priority is given to the interrupted SGUs to decrease the transmission delay. If an emergency data from any class SGU gets interrupted, it gets the highest priority. The emergency interrupted data of class c2 has higher priority than the interrupted data of class c1.

In summary, there are a total of five queues based on assigned priorities. From high to low, these priority queues are:

1) Primary Users (PUs), being licensed users
2) Interrupted emergency SGUs of any class
3) Interrupted SGUs
4) Emergency SGUs
5) Newly coming SGUs

Using the above-mentioned priority policy, the scheduler assigns available channels to SGUs present in high priority queues. The data transmission from low priority SGUs is interrupted on the arrival of PUs or high priority SGUs. These interrupted SGUs will be adjusted in the their queues to wait for the channel re-assignment. The traffic scheduler will assign the available channels to SGUs in each time slot to minimize the transmission delay of SGUs. The transmission scheduling problem is divided into number of stages. Some key components of the schedular are described below. The detailed description can be found in [5].

A. Stage

The complete scheduling process is divided into stages with $\Delta\tau$ being the duration of each stage. The channel allocation decision is made at the start of each stage.

B. State

The system state $\mathbf{s}(k)$ at the k^{th} stage is defined by

$$\mathbf{s}(k) = (ch_n(k), s_m(k) | n = 1, ..., N, m = 1, ..., M) \quad (1)$$

where $ch_n(k)$ is the availability of channel n at stage k, $s_m(k)$ is the state of m^{th} SGU at stage k which include the number of packets contained by the SGU, interrupt and emergency flags, N is the total number of PUs (same as total number of channels), and M is the total number of SGUs.

C. Decision

The decision at k^{th} stage is given by

$$\gamma(k) = (\gamma_m(k) | m = 1, ..., M) \quad (2)$$

where $\gamma_m(k)$ is a number from $1, ..., N$ indicating the channel number assigned to m^{th} SGU during k^{th} stage. The complete decision space is $\mathbf{\Gamma}$ which contains all the possible decisions against system state $\mathbf{s}(k)$.

D. Decision Policy

Decision policy is defined as functions to decide sequential allocation of channels. It is the mapping from system state to the decision vector.

E. Utility Function

The utility function at stage k as a function of the system state and decision, and describing the overall transmission delay is given as

$$\Gamma[\mathbf{s}(k), \gamma(k)] = \sum_{m=1}^{M} \omega_m(k)\tau_m(k)\mathbf{1}_{p_m(k)>0} \quad (3)$$

where $\omega_m(k)$ is the weight of m^{th} SGU based on its role class (either c1 or c2), emergency, and interrupt data (see [5]), $\tau_m(k)$ is the tranmission delay of m^{th} SGU which is equal to $\Delta\tau$ if an SGU is unable to get a channel, and 0 otherwise, $\mathbf{1}_{p_m(k)>0}$ is the indicator function conditioned on the number of non-zero packets contained by m^{th} SGU.

F. System Cost

System cost is defined as the overall transmission delay of the entire scheduling process. Under a decision policy μ, the system cost is provided by

$$J[\mathbf{s}(k)] = \sum_{j=k}^{\infty} \Gamma(\mathbf{s}(j), \mu[\mathbf{s}(k)]) \quad (4)$$

The aim of scheduling process is to produce the optimal decision policy to minimize the system cost, leading to minimum system cost.

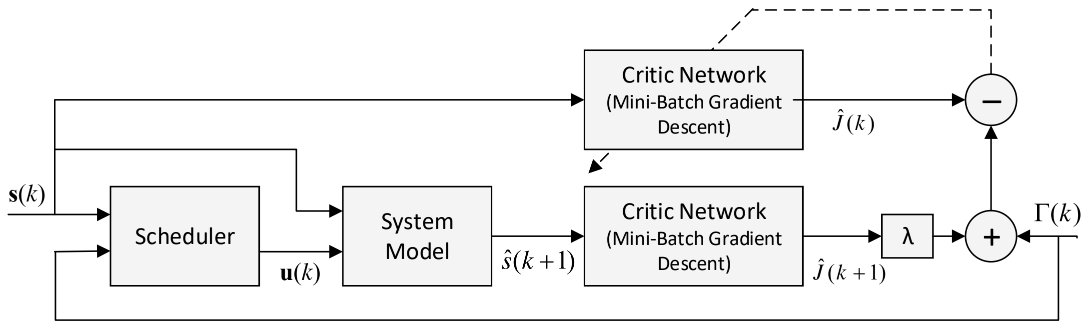

Fig. 1. Proposed HDP Architecture using Mini-batch Gradient Descent Optimization

IV. Proposed Optimization using Mini-batch Gradient Descent

In [5], Adaptive Dynamic Programming (ADP) approach is introduced to solve the traffic scheduling problem through optimization via critic network. Specifically, Heuristic Dynamic Programming (HDP) is adopted which typically comprises of action (scheduler), system model and critic networks [11] (see figure 1). In [5], gradient descent method is used in the critic network for minimization of system cost. In this paper, we propose novel usage of mini-batch gradient descent optimizer for the minimization of the system cost. Mini-batch gradient is selected due to optimized speed of learning and processing time as compared to stochastic gradient descent which takes more time to learn for large training set and batch gradient descent which take more time for computation [12]. The critic network of the system is based on Neural Network, which is trained offline using backpropagation. The offline training of neural network is more effective and helps to choose the hyperparameters more efficiently. Moreover, a trained network takes less time to execute and predict the output. The critic network architecture consists of two layers, one hidden layer and an output layer, as shown in figure 2.

The activation function $g(Z_h)$ for hidden layer is sigmoid function given by

$$g(Z_h) = \frac{1 - e^{Z_h}}{1 + e^{Z_h}} \tag{5}$$

where Z_h is input of sigmoid function, $g(Z_h)$ is the output of activation units of hidden layers, and

$$Z_h = W_h X \tag{6}$$

where W_h are weights of hidden layers, and X is input of hidden layer/Critic Network. The activation function for output layer is linear function given by

$$Y = W_o g(Z_h) \tag{7}$$

where Y is the output of activation unit of output layer, W_0 are the weights for output layer, and $g(Z_h)$ is the input of the output layer.

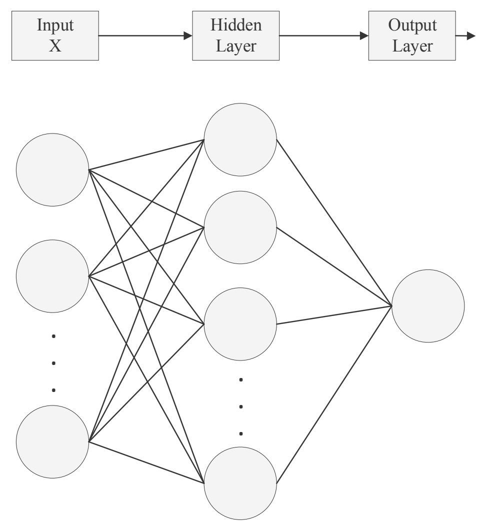

Fig. 2. Critic Network Architecture

The mean square function $E(W)$ (or the loss/error function) for our neural network is given by

$$E(W) = \sum_k [\hat{J}(k) - \Gamma(k) - \lambda \hat{J}(k+1)]^2 \tag{8}$$

where W is weight on which $E(W)$ is computed, λ is the discount factor $(0 \leq \lambda \leq 1)$, k is the stage number or example set on which error is calculated, $\hat{J}(k)$ is predicted value system cost for k^{th} stage, $\Gamma(k) + \lambda\hat{J}(k+1)$ is computed value using forward propagation for k^{th} stage. The weight updates according to mini-batch gradient descent method are given by

$$W = W - \alpha \Delta W \tag{9}$$

$$\Delta W = \frac{\partial E(W)}{\partial W} \tag{10}$$

where α is learning rate for neural network, ΔW is the change of weight with respect to error function $E(W)$. Equations 9 and 10 describe the mini-batch gradient descent method for optimization of $E(W)$ which updates the value of W for each E.

For stochastic gradient descent, W is updated for each example of training set which increases the uncertainty in optimization and it may stuck in local minima. In other words, it is difficult to find the global minima in stochastic gradient descent optimization technique. For mini-batch gradient descent optimization, W is updated after computing the $E(W)$ for all of examples in training set. For mini-batch gradient descent optimization, training set is divided into small chunks called mini batches. The value of k in eq. 8 is equal to the size of mini-batch. Mini-batch size is normally chosen as 2^q where q is positive integer based upon requirement. For same number of iteration, stochastic gradient descent is uncertain to find the global minima, while mini-batch gradient descent takes more time to train the neural network. The mini-batch gradient descent is trade off between computation power and training time.

V. Simulation Results

In simulations, we have considered as CR-based SG network with number of PUs $N = 4$, and number of SGUs $M = 8$. According to the selected 2-class model, these SGUs can belong to either class c1 or c2. In our simulation model, SGUs 1-4 belong to c1, and SGUs 5-8 belong to c2. The duration of each stage $\Delta\tau$ is taken as unity. Both the PUs and SGUs arrive according to poisson process with arrival rates of λ_p and λ_s, respectively. The input of critic network is a column vector of length 28 which consists of four channel states, and 8 SGUs' states. The hyperparameters which includes number of hidden layers, number of activation units in each layer, size of mini-batch and number of iterations to run are determined such that the critic network should not be high bias or high variance.

The size of 64 is selected for mini-batch. The weights for critic network are initialized randomly. There are 500 iterates and 35 hidden layer activation units with learning rate of 0.2. The dimension of W_h is 28x35. The critic network is implement in vector form which results in Z_h as a column vector of length 35. The dimension for W_0 is 35x1 which generates $\hat{J}$ as scalar value for each stage. There are total 6000 examples which are divided into two subset training set and test set with 5000 and 1000 examples, respectively. The examples are collected by implementing and executing the traffic schedular in MATLAB. Critic network is trained offline using backpropagation and weights are updated via mini-batch gradient descent optimizer. The critic network is trained on open-sources Keras framework in python.

Figure 3 shows the comparison of system costs for the stochastic gradient descent and mini-batch gradient descent optimizers plotted versus number of iterations. In this simulation, batch size of 64 and learning rate of 0.2 are used. In this figure, first 1600 iterates are plotted for better comprehension. It is clear that proposed mini-batch gradient descent optimizer results in relatively smooth system cost due to the use of averaging. Another reason is that the global minima is found relatively faster as compared to stochastic gradient descent method which often stucks in local minima.

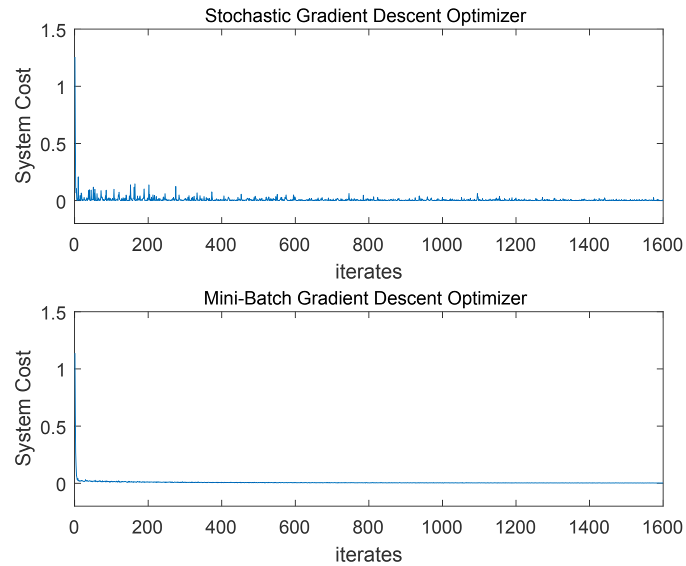

Fig. 3. Comparison of system cost for the stochastic gradient descent and mini-batch gradient descent optimizers

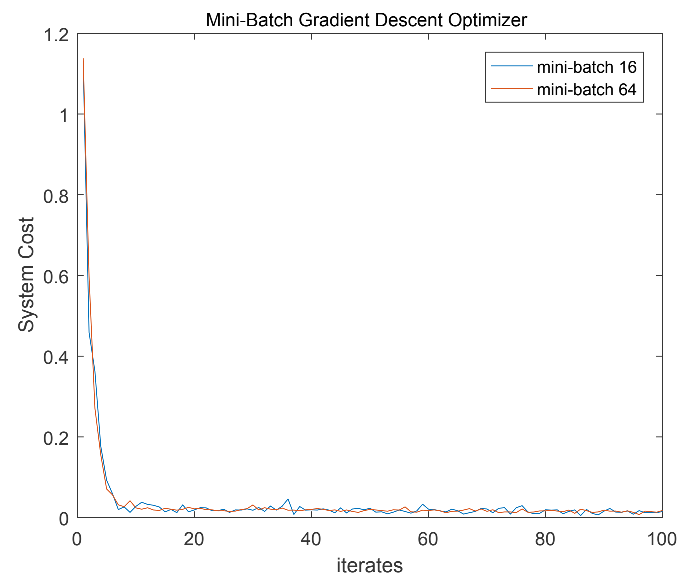

Fig. 4. System cost for mini-batch gradient descent optimizers for batch sizes of 16 and 64

Figure 4 shows the comparison of system costs for mini-batch gradient descent optimizers for batch sizes of 16 and 64 are plotted versus number of iterations. Learning rate of 0.2 is used in this simulation. Figure shows that batch size of 64 results in smoother cost function as compared to the batch size of 16. Both these results demonstrate that the proposed novel usage of mini-batch gradient descent optimizer for the traffic scheduling of CR-based SG network results in better optimization performance as compared to the stochastic gradient method.

VI. CONCLUSION

Smart Grid (SG) network consists of several applications and multiple network layers resulting in differential QoS requirement in terms of latency, throughput and reliability. Cognitive Radio (CR), which offers better spectrum utilization through opportunistic spectrum access and spectrum sharing, is considered as a promising wireless technology for SG networks. Traffic scheduling and optimization is one of major challenges in CR-based SG communication network. In this paper, a novel usage of mini-batch gradient descent method for the optimization of QoS-based differential scheduling in CR-based SG network is proposed. A 2-class based priority scheduling model with emergency and interrupt handling capability is used. Simulation results have shown that mini-batch gradient descent method achieves better results as compared to gradient descent method in terms of the fast convergence and minimization of the overall cost function.

REFERENCES

[1] A. A. Khan, M. H. Rehmani, and M. Reisslein, "Cognitive radio for smart grids: Survey of architectures, spectrum sensing mechanisms, and networking protocols," *IEEE Communication Surveys and Tutorials*, vol. 18, no. 1, pp. 860–898, 2016.

[2] A. O. Otuoze, M. W. Mustafa, and R. M. Larik, "Smart grids security challenges: Classification by sources of threats, in press," *Journal of Electrical Systems and Information Technology*, 2018.

[3] C. Qian, Z. Luo, X. Tian, X. Wang, and M. Guizani, "Cognitive transmission based on data priority classification in WSNs for smart grid," in *Proc. of IEEE GLOBECOMM*, 2012, pp. 5166–5171.

[4] J. Zhou, R. Q. Hu, and Y. Qian, "Traffic scheduling for smart grid in rural areas with cognitive radios," in *Proc. of IEEE GLOBECOMM*, 2012, pp. 5172–5176.

[5] R. Yu, W. Zhong, S. Xie, Y. Zhang, and Y. Zhang, "QoS differential scheduling in cognitive-radio-based smart grid networks: An adaptive dynamic programming approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 435–443, 2016.

[6] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in HAN, NAN and WAN," *Computer Networks*, vol. 67, pp. 74–88, 2014.

[7] R. Ranganathan, R. Qiu, Z. Hu, S. Hou, M. Pazos-Revilla, G. Zheng, Z. Chen, and N. Guo, "Cognitive radio for smart grid: Theory, algorithms, and security," *International Journal of Digital Multimedia Broadcasting*, pp. 1–15, 2011.

[8] A. Cacciapuoti, M. Caleffi, F. Marino, and L. Paura, "Sensing-time optimization in cognitive radio enabling smart grid," in *Proc. of EMTC*, 2014, pp. 1–6.

[9] V. C. Gungor and D. Sahin, "Cognitive radio network for smart grid applications," *IEEE Vehicular Technology Magazine*, vol. 7, no. 2, pp. 41–46, 2012.

[10] "Communication requirements of smart grid technologies," US Department of Energy, Tech. Rep., 2010.

[11] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Computational Intelligence Magazine.*, vol. 4, no. 2, pp. 39–47, 2009.

[12] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Mini-batch gradient descent: Faster convergence under data sparsity," in *Proc. of the IEEE 56th Annual Conference on Decision and Control*, 2017, pp. 2880–2887.