Optimized Transfer Learning: Application for Wireless Channel Selection

Mohammad Askarizadeh*‡, Mostafa Hussien*, Masoumeh Zare†, and Kim Khoa Nguyen*
* Dept. Electrical Engineering, ÉTS, Université du Québec, Montréal, QC, Canada
† Dept. Economics, Université de Montréal, Montréal, QC, Canada
‡ mohammad.askarizadeh-khanaman.1@ens.etsmtl.ca

***Abstract*—Recently, transfer learning (TL) has emerged as a powerful machine learning method in distributed environments. Transferring the knowledge between distributed agents helps reduce both learning time and computing costs. However, in a communication system, the advantage of TL comes with communication costs. To make an optimal decision of transfer between two agents, we try to answer three key questions: i) which information should be transferred from a source to a target?, ii) how this transferred information will be adapted to the target? and iii) when should TL be triggered to optimize the costs?. To this end, we introduce a new concept of similarity based on the *Best Approximation Theory* and a general transfer rule. Then, we propose a model to evaluate the feasibility and optimality of TL. We verify our proposed model in the context of the wireless channel selection problem using contextual multi-armed bandits. Experimental results show optimal TL decisions can be made, and *Extra Action* is an efficient technique for TL in channel selection.**

***Index Terms*—Transfer learning, optimization, similarity, channel selection, contextual multi-armed bandit.**

I. Introduction

Although traditional machine learning (ML) techniques, such as deep learning (DL) and reinforcement learning (RL), have achieved outstanding results in various domains. A major problem of DL is it requires huge datasets, which is often costly or impractical in several situations [1]. Another issue arises in RL techniques is the massive sample complexity of RL methods. Transfer learning (TL) is a promising ML technique for tackling these critical challenges thanks to its unique ability of transferring information across different domains. The most instinctive approach to utilize TL for RL is to reuse the answer of prior tasks that have already been completed [2].

Despite their considerable advantages, no prior work has yet successfully modeled or optimized TL [2], [3]. Indeed, an efficient application of TL should respond to three following questions: what should be transferred from a source to a target?, and how this transferred information should be adapted in the target task?, In addition, unlike the centralized machine learning approach, TL consumes communication resources. Therefore, an additional question should be considered in networked environments: when should TL be triggered to optimize the costs?.

Attempts to answer the first two questions have been presented in prior work. Determining the information to transfer is a challenging issue, because transferable information can be changed over time, and from one problem to another. Different kinds of transferable information have been suggested such as Value Function, Reward Function, Policy, Rule, Action Set, etc. [2]. For instance, In [4], the authors use reward function obtained from the preceding task or another agent, to have a more informative reward function. In [5], a portion of the policy is transferred while in several other works an appropriate subset of accessible actions is found and transferred to another task to accelerate learning [6], [7].

To clarify how transferred information should be adapted in the target task, first it ought to be specified how this information should be selected from the source. To answer this question, various similarity concepts have been considered. This concept can be clearly viewed in the context of classical metrics. Some works have introduced a similarity concept according to their especial problems, without following classical metric, defining similarity manipulately [8]. We introduce a new generally applicable similarity concept based on the Best Approximation Theory.

On the other hand, transferred information is not usually useful for the target task. Therefore, prior work has investigated how transferable information can be adapted to a target task. In [6], [7], authors suggest a platform called giving advice, and actions play the role of advice from advisor to advisee. The time of giving advice is when the advisee is not sure what to do. An idea that is similar to [6], [7] named student-teacher is suggested by [9]. Indeed, value functions are considered as transferred information in [9]. Some works introduce a notion called transfer rule. Indeed, transfer rules work as dictionaries. For example, [10] recommends a technique to learn a transfer rule automatically from contacts with the environment. In this paper, we also present a new transfer rule based on our proposed similarity.

To the best of our knowledge, this work is the first attempt to answer the third question by modelling the trade-off between the accuracy-gain and cost of TL. Our proposed model is composed of two elements: 1) the cost of TL in terms of communication and computation time, and 2) the learning advantage of TL in terms of the employed performance metric (e.g., cumulative reward, accuracy, loss, etc). We propose a new utility and optimization model based on an economical notion for taking advantage of an optimized TL.

Indeed, RL has recently become a preferred solution for various wireless communication problems such as beam se-

978-3-903176-36-2 ©2021 IFIP

lection, link adaptation, or channel selection [11]. To improve the performance of RL in distributed scenarios, TL can be an efficient candidate. In communication systems, the advantage of TL comes with computation and communication costs. Therefore, the decision of transferring the knowledge between agents should be optimized. According to this motivation, we verify our mathematical model in the context of a channel selection problem using a contextual multi armed bandits (CMAB) agent. Indeed, CMAB has been adopted in many prior works to cope with the key challenge facing any channel selection technique, namely the dynamic nature of wireless channels [12]. In this scenario, the pretrained CMAB agent resides in a base station (BS) and its learned information needs to be transferred to another BS through a control channel, for improving the quality of channel selection by using TL in this BS. Moreover, our proposed utility and optimization model is applied in channel selection problem to decide whether we should use TL in a given case or not and when TL is optimal.

The main contributions of this work can be summarized as follows:

- Proposing a new general similarity concept and a transfer rule that constitute an efficient framework to acquire learned knowledge from a source task to a target task.
- Introducing a utility and optimization model to determine when TL should be adopted, and making use of this model in the case of the channel selection problem.
- Improving the performance of channel selection problem using our proposed concepts and showing when these improvements are feasible and optimal.

II. Transfer Learning

Transfer learning (TL) can be divided into different categories regarding RL. Basically, the process of the TL techniques used in this paper can be summarized as follows:

1) Learn an optimal policy in the source task and recording a two-column matrix S_A of experience information (states, actions) by following the learned policy.
2) Translate the experience matrix using a suitable transfer rule, being understandable for target task. The transfer rule explained in section III is applied to the output from the first step.
3) Incorporate the learned knowledge: In this step, we incorporate the output of the previous steps (output of the transfer rule) in the training of the target task. This step is slightly different based on the selected TL method. Generally, it consists of two phases. In the first phase, a Deep Q-learning is updated for some epochs with actions recommended by the transfer rule. In the second phase, the network is updated according to the normal training process (training the target with its original RL technique).

III. Similarity and Transfer Rule

To apply TL in any problem, a number of diverse factors should be precisely defined. One of these factors is defining a suitable transfer rule (translation function) to translate the knowledge obtained in the source task to a form that can be useful in the target task. Our proposed transfer rule is based on the similarity concept inspired by the *Best Approximation Theory* [13].

A. Best Approximation and Similarity

Starting with a short background on the *Best Approximation Theory*, let X be an inner product space ($n-$dimensional Euclidean space), and C be a subset of this inner product space. Element p in subset C is called the best approximation to x, which is an element of inner product space X, if p is the closest element to x among other members of subset C, it means

$$\|x - p\| = d\,(x, C)\,, \tag{1}$$

where

$$d\,(x, C) := \inf\{\|x - y\| \mid y \in C\}. \tag{2}$$

The set of all best approximations x to C is indicated by $P_C(x)$, which is:

$$P_C(x) := \{y \in C \mid \|x - y\| = d\,(x, C)\}, \tag{3}$$

means that the most similar element of C to x belongs to $P_C(x)$.

B. Transfer Rule

Having the similarity concept defined, state and action spaces of source and target tasks are denoted by S_1, A_1, and S_2, A_2, respectively. The training of the source task is assumed to be completed before starting the transfer process. Now, the best action for a given state in the target task should be predicted based on the defined transfer rule. Without loss of generality, a decision is made to find the best action for only one state in the target task (s_2). The recommended transfer rule is defined as follows:

$$\pi_2(s_2) := P_{A_2}(\pi_1(P_{S_1}(s_2))), \tag{4}$$

where s_2 is an arbitrary state in S_2, P is best approximation function defined in equation (3), and π_1, π_2 are the best policy in source and target, respectively.

IV. Optimized Transfer Learning

In this section, we discuss the question of when using a TL is cost-effective and optimal by considering its cost. In order to explore the economical and optimality aspects of TL, it is necessary to define the utility of an agent. Then, according to this definition, exploring an optimized TL is possible.

A. Communication and Computation Cost

To evaluate the feasibility of a learning method, computation, communication time, and energy consumption are widely considered in the literature as cost of learning [14]. We assume BSs have unlimited power resources and therefore we ignore the energy consumption cost. We focus on computation and communication time as the cost of TL. In addition, the computation time required to retrieve this information from

the source task participates in computing the total cost. On the other hand, the cost of training the target task without TL is only the computation time required for training the agent from scratch.

1) Computation Time: We consider the execution time of the model as computation cost, assuming an equal computational power at each BS [15]. Moreover, we assume a Deep Q-learning with L fully connected layers is used to approximate the Q-function in the TL technique. So, given these assumptions, the computation cost is given by:

$$T_{cop_{TL}} = \mathcal{O}(|S_A|) + \mathcal{O}(|S_A|(i_1 i_2 + \cdots + i_{L-1} i_L)) + \mathcal{O}(n(i_1 i_2 + \cdots + i_{L-1} i_L)), \quad (5)$$

where $i_1, i_2, ..., i_L$ are the number of nodes in the Deep Q-learning layers. $|S_A|$ is the size of selected sample, the number of rows of the experience matrix, S_A, transferred from the source task, and n is the number of epochs. Moreover, in (5), the first term is the execution time of the computation resource, the second and third terms are the execution time of the first and second phases of the TL network respectively.

2) Communication Time: The communication time considered here is the time required for transmitting the learned information from the source to the target task in a TL model. The transmission rate is defined as follows [14]:

$$r = B \ln(1 + \frac{hp}{N}), \quad (6)$$

where B is the channel bandwidth, N is the noise power, p is the transmission power, and h is the channel gain. Therefore, communication time allocated for transmitting the learned information, S_A, to the target is:

$$T_{com} = \frac{c \times |S_A|}{r}, \quad (7)$$

where c is a constant of converting the matrix information to the number of bits.

B. Utility and Optimization

Having introduced the computation and communication cost of TL, in this section, we introduce the mathematical model of an agent's utility. Inspired from the economical notion, we define the utility of an agent to be income minus cost. Generally, one of the performance metrics which are selected to evaluate the learning task can be employed as income for TL process. Examples of these metrics are Jump-Start, Asymptotic Performance, Accumulated Reward (Total Reward), Transfer Ratio, or Time to Threshold [2], [8]. Choosing one of these metrics (M), the utility of an agent is defined as follows:

$$U_{TL} = M_{TL} - T_{com} - T_{cop_{TL}}, \quad (8)$$

while the utility of the learning without TL is:

$$U_{RL} = M_{RL} - T_{cop_{RL}}, \quad (9)$$

where $T_{cop_{RL}}$ is exactly the third part in (5). According to the aforementioned utility model, using a TL technique to improve the learning of a target task is feasible whenever $U_{TL} > U_{RL}$. This inequality is interpreted as a feasibility condition indicating whether using TL is a cost-effective. Therefore, by setting special fixed parameters sample size, rate of communication, and network structure, it can be specified when using TL is economical. Furthermore, to figure out the optimal sample size and optimal utility value, it is necessary to define an optimization model.

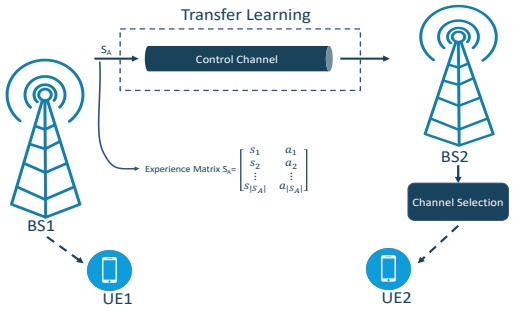

Fig. 1. Transfer of learned information between two base stations.

Now, we present an optimization model to maximize the agent's utility. By setting the sample size to x i.e., $|S_A| = x$, the optimization model is defined as:

$$\max_x U_{TL}(x) = \max_x (M_{TL}(x) - x(\frac{c}{r} + b)), \quad (10)$$

where, M_{TL} is selected performance metric, c is defined in (7), $b = \mathcal{O}(i_1 i_2 + \cdots + i_{L-1} i_L)$. Regarding this optimization model, setting a fixed communication rate and network structure, the maximum utility of TL is identified in terms of optimal sample size.

V. Channel Selection

We validate our mathematical model in the context of channel selection problem. Indeed, for modeling channel selection problem, we consider CMAB model selected by many prior work to tackle with the dynamically changing wireless channels. We use discrete contextual bandits (DCB) algorithm to select actions. DCB method is a powerful generalization of the widely used Upper Confidence Bound algorithm (UCB) [12], [16]. In this scenario, the pretrained CMAB agent resides at the BS1 communicating with UE1. In fact, BS1 learns channel selection for UE1 for a typical uplink channel selection problem. A two-column matrix learned information, S_A, needs to be transferred to the BS2 through a special control channel. BS2 improves the quality of channel selection which is learning for UE2 in the case of another typical uplink channel selection problem, see Fig. 1.

At each time step, a packet of length k bits should be transmitted where k is randomly selected from a set of predefined packet lengths. In CMAB setting, the packet length is thought of as the current context and the different channels represent the arms to be pulled. A reward function specifies the success or failure of transmission by delivering values between the numbers of available channels. A Deep Q-learning

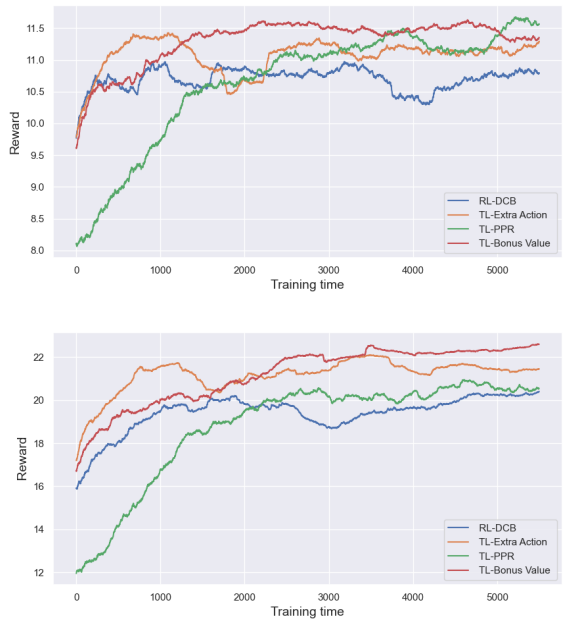

Fig. 2. The expected reward for a channel selection problem with 13 (top) and 24 (down) channels.

is employed to approximate the Q-value of each action given the current context.

We employ two performance metrics namely *Accumulated Reward* (AR), and *Transfer Ratio* (TR). AR is the area under the curve of rewards received during the training process. The TR is the ratio between the AR's received by the two techniques to be compared (e.g., with and without transfer).

VI. Experimental Results

A. Implementation Details

We setup a channel selection problem as a source task with $K = 12$ different packet lengths and $N = 12$ available transmission channels. Moreover, it is assumed that the packet lengths at the source and target tasks are drawn from the same distribution. We train both the source and target tasks for 6000 epochs. We use a Deep Q-learning with one hidden layer of 100 nodes. The number of nodes in the input layer equals the number of different data bits, while the number of nodes in the output layer is equal to the number of available channels (i.e., actions). Additionally, we use ReLU activation in the hidden layer and linear activation in the output layer. Furthermore, a learning rate of $1e-2$ is used with *Adam* optimizer to train the models in the source and target tasks.

B. Results

We evaluate the performance of the proposed model using six different experiments. In the first experiment, we test various applied TL methods, namely *Extra Action*, *Value Bonus*, and *Probabilistic Policy Reuse (PPR)* [8] in the channel selection problem. The obtained performance is compared to this of the CMAB trained using DCB algorithm. The same experiment is repeated using different number of channels to choose from, namely 13 and 24. We test two various number of channels to validate our model in two cases where the number of available channels is close and far from the available channels in the source task. Fig. 2 illustrates the results of these experiments. As shown in Fig. 2, all TL methods obtained better reward compared with DCB without TL. Sample selections $|S_A| = 200, 500$ are chosen for cases $N = 13, 24$ respectively. Moreover, for both experiments, we add a value of 5 as an added bonus value for testing *Value Bonus* technique. While in PPR we employed the following parameters $\epsilon = 0.1$, and $\psi = 0.999$. Table I presents the results of the second experiment. This experiment illustrates the results of testing *Extra Action* method with various number of sample sizes namely, $500, 600, 700$, and 800. The number of available channels in the target task is set to $N = 24$. Indeed, two metrics AR, and TR are employed in this experiment. Note that the AR is computed according to the following equation:

$$AR = \sum_{t=1}^{epochs} \text{average reward at time } t. \tag{11}$$

As shown in table I, both metrics have shown a great improvement compared with prior results of DCB. The third experiment evaluates the feasibility aspect of TL, checking the inequality $U_{TL} > U_{RL}$, see Fig. 3. To obtain comparable values, all parameters in equations (8) and (9) are normalized. A new parameter, $\alpha \in [0, 1]$, is defined to practically control (8). Regrading α, equation (8) changes as follows:

$$U_{TL} = AR_{TL} - \alpha(T_{com} + T_{cop_{TL}}). \tag{12}$$

Setting a value for α in various situations depends on the cost-importance to the agent. The agent in this experiment has 24 available channels and use *Extra Action* method. Some parameters for cheking the feasibility are calculated and fixed as follows, $c = 64$, $b = 3600$, $\alpha = 0.1084$, and $|S_A| = 1025$. Moreover, for calculating r, we use following relation:

$$r = B \log(1 + \text{SNR}), \tag{13}$$

where $B = 20 \times 10^6$. By changing the value of SNR, we can decide when TL is feasible. In fact, according to Fig. 3, for SNRs more than $8dB$, using TL is feasible. This conclusion can be explained by the fact that when the control channel suffers bad propagation conditions, the cost incurred in TL is more than its income and it is not worth to apply TL in this case.

In addition, Table II shows results of testing *Value Bonus* method with different bonus values. Typically, we test bonus values of $1, 5, 10$, and 15. In this case, we fix the number of channels to $N = 24$ and the sample size to $|S_A| = 500$. Note that the last column shows the upper bound value of α. Finally, Fig. 4 shows an evaluation for the optimization model. This experiment uses *Extra Action* method with SNR$= 30$. Since in this use case, we do not have a close form of AR_{TL}, we approximate it with a second degree polynomial. Therefore, this

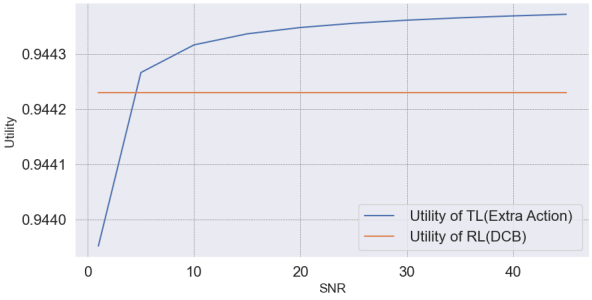

Fig. 3. Agent's utility with $\alpha = 0.1084$ and different SNR.

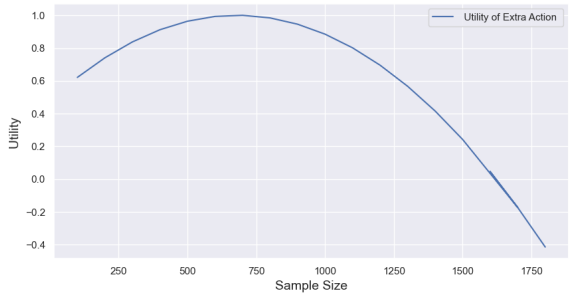

Fig. 4. Agent's utility for different sample sizes with SNR= 30 dB.

evaluation of optimization model is approximately. According to the Fig. 4, the optimal sample size is $|S_A| = 675$.

TABLE I
TRANSFER LEARNING FROM 12-CHANNELS TO 24-CHANNELS AGENTS.

Selected Samples	Accumulated Reward		Transfer Ratio
	Extra Action	*DCB*	
500	125776	115544	0.919
600	127240	122239	0.961
700	127767	118063	0.924
800	128518	123767	0.963

TABLE II
ALPHA UPPER BOUND FOR *Value Bonus* WITH N=24 AND $|S_A|$=500.

Bonus	Accumulated Reward		Transfer Ratio	UB-Alpha
	Value Bonus	*DCB*		
1	128179	122972	0.9593	0.162
5	127342	122972	0.9656	0.135
10	129115	122972	0.9524	0.19
15	128556	122972	0.9566	0.174

VII. CONCLUSION

In this work, a new general similarity concept and transfer rule have been introduced. Then, a utility concept and an optimization model of transfer learning were suggested. Using the proposed framework, one can decide when it is feasible and optimal to apply transfer learning and when it would be better to learn from scratch. Although applied in channel selection problem, these contributions are not limited for this problem and they can be easily extended to different contexts. Numerical results showed that using *Extra Action* obtained a 91% transfer ratio.

ACKNOWLEDGMENT

The authors thank Mitacs and Ciena for supporting this research in the IT13947 grant.

REFERENCES

[1] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[2] F. L. Da Silva and A. H. R. Costa, "A survey on transfer learning for multiagent reinforcement learning systems," *Journal of Artificial Intelligence Research*, vol. 64, pp. 645–703, 2019.

[3] C. T. Nguyen, N. Van Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q.-V. Pham, D. Niyato, E. Dutkiewicz, and W.-J. Hwang, "Transfer learning for future wireless networks: A comprehensive survey," *arXiv preprint arXiv:2102.07572*, 2021.

[4] H. B. Suay, T. Brys, M. E. Taylor, and S. Chernova, "Learning from demonstration for shaping through inverse reinforcement learning," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 2016, pp. 429–437.

[5] T. Sakato, M. Ozeki, and N. Oka, "Learning through imitation and reinforcement learning: Toward the acquisition of painting motions," in *2014 IIAI 3rd International Conference on Advanced Applied Informatics*. IEEE, 2014, pp. 873–880.

[6] S. Omidshafiei, D.-K. Kim, M. Liu, G. Tesauro, M. Riemer, C. Amato, M. Campbell, and J. P. How, "Learning to teach in cooperative multiagent reinforcement learning," in *AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6128–6136.

[7] A. Fachantidis, M. E. Taylor, and I. Vlahavas, "Learning to teach reinforcement learning agents," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 21–42, 2019.

[8] M. E. Taylor and P. Stone, "Cross-domain transfer for reinforcement learning," in *International conference on Machine learning (ICML)*, 2007, pp. 879–886.

[9] H. Kono, A. Kamimura, K. Tomita, and T. Suzuki, "Transfer learning method using ontology for heterogeneous multi-agent reinforcement learning," *International Journal of Advanced Computer Science & Applications*, vol. 5, no. 10, 2014.

[10] T. Croonenborghs, K. Driessens, and M. Bruynooghe, "Learning a transfer function for reinforcement learning problems," in *AAAI Workshop on Transfer Learning for Complex Tasks*. AAAI Press; Menlo Park, California 94025, USA, 2008, pp. 1–6.

[11] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[12] P. Zhou, J. Xu, W. Wang, C. Jiang, K. Wang, and J. Hu, "Human-behavior and qoe-aware dynamic channel allocation for 5g networks: A latent contextual bandit learning approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 436–451, 2020.

[13] F. R. Deutsch, *Best approximation in inner product spaces*. Springer Science & Business Media, 2012.

[14] C. T. Dinh, N. H. Tran, M. N. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Transactions on Networking*, 2020.

[15] C. Ma, J. Konečnỳ, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takáč, "Distributed optimization with arbitrary local solvers," *Optimization Methods and Software*, vol. 32, no. 4, pp. 813–848, 2017.

[16] P. Sakulkar and B. Krishnamachari, "Stochastic contextual bandits with known reward functions," *arXiv preprint arXiv:1605.00176*, 2016.