Optimal Crowdsourced Channel Monitoring in Cognitive Radio Networks

Ahmed M. Salama[1], Ming Li[1], Dejun Yang[2]
[1] Dept. of ECE, The University of Arizona
[2] Dept. of EECS, Colorado School of Mines
E-mail{ahmedsalama,lim}@email.arizona.edu, djyang@mines.edu

***Abstract*—Crowdsourcing is an emerging paradigm for spectrum access rule enforcement in dynamic spectrum sharing, which leverages a large number of mobile users to help monitoring and detecting spectrum violations and misuse. Its main advantages compared with traditional dedicated monitoring architecture includes enhanced coverage, effectiveness and lower costs. However, how to optimally assign mobile users to monitor the channel usage has not been studied in the crowdsourced setting. The main challenges are: the large number of channels to monitor while mobile users may not be available all the time, the need to consider monitoring costs and incentives, as well as the uncertainty of each channel's traffic patterns. In this paper, we tackle such challenges by formulating a stochastic optimization problem that optimizes the spectrum monitoring task for crowdsourced mobile users. We consider the availability pattern of the mobile users and we assume they are given payments as incentives for participating in monitoring. Simulations show that our method outperforms the risk-averse scenario and has a small gap with the solution under perfect information.**

I. Introduction

The huge demand for high data rates, the scarcity of the legitimate spectrum offered by the spectrum providers and the increasing number of mobile users, made efficient spectrum utilization an undeniable need. Recent studies [1] show that the existing spectrum assignment policies are underutilizing the licensed band. Hence, cognitive radio [2] is proposed to maximize the use of the spectrum holes vacated by the primary users (PUs). Although utilizing those vacancies can be permitted by a main control hub or a *platform* for a certain time window and certain secondary users (SUs), some SUs could access the spectrum without authorization causing interference to both of the PUs and the authorized SUs [3]. Therefore, monitoring the spectrum to detect those violations is a necessary action. However, the traditional approach to spectrum monitoring relies on a set of dedicated monitors, which suffer from several drawbacks: (1) The large number of channels makes it hard to cover all the channels, which affects the monitoring coverage and detection effectiveness; (2) With a small number of monitors, it is difficult to estimate and predict the traffic pattern accurately, while frequent switching among different channels incurs packet losses; (3) The cost of deployment is high.

To solve these problems, *crowdsourcing* [4], [5] spectrum monitoring approaches that leverage the power of a crowd of mobile users have been proposed to distribute the monitoring burden and to accumulate more accurate results. Although the channel assignment problem may seem simpler as we have more monitors, the optimal assignment is still challenging. Specifically, we need to consider the constraints on the availability period of each mobile user, as well as provide incentives to users to encourage their participation in monitoring, otherwise, mobile users would waste their communication time and energy. In addition, the platform needs to minimize its total cost which includes the payment to all the monitors. Finally, the traffic pattern on each channel is not deterministic and may not be known ahead of time, which adds uncertainty to the monitoring result. The size of the problem (large number of monitors and channels) also potentially make the solution computationally intractable.

The problem of channel assignment has been studied under non-crowdsourcing based models. In [6], the authors propose a greedy algorithm to maximize packet collection with the fewest channel switches, but this algorithm is not proven to be optimal. In [7], the authors formulate an NP- hard ILP that maximizes the number of active slots captured and they use the probabilistic rounding algorithm (PRA) to solve the LP-relaxation of the ILP. Similar to [7], in [8], the authors employ PRA and compare it with a deterministic rounding algorithm (DRA) and they prove that DRA outperforms PRA in achieving the best rounding ratio. In [9], using PRA as well, the authors formulate and solve the assignment problem given the knowledge of the transmission probability of the monitored nodes. Although these works efficiently approximate the solution of the NP-hard assignment problem by rounding, the solutions they deliver are not optimal and does not consider the monitors' incentives. On the other hand, a few crowdsourcing based mechanisms considered incentivizing the monitors. In [10], the authors design an auction-game based incentive mechanism, while formulating a dynamic program that minimizes the total cost of monitoring. In [11], they consider a more general model by including service requesters as players. Although these works properly reward the monitors, they do not consider an optimal channel assignment problem.

In this work, we propose an optimal channel assignment framework for crowdsourcing-based spectrum monitoring. We consider the monitors as mobile users having an availability pattern that represents their busy and free times. We not only dedicate monitors to channels in the frequency domain, but also, we assign them to the channels over T time slots in the future. Due to the monitors' activity patterns, they are not available all the time to monitor. This forces us to assign them channels to monitor in their free times only. We deal with the traffic pattern in two ways. First, we assume a deterministic traffic pattern (DTP), where the number of packets arriving in a certain time on a specific channel is known. Second, we assume a stochastic traffic pattern (STP) where the traffic pattern is a random variable (RV) with known distribution. We assume that the platform gives incentive to each monitor depending on the fixed sensing cost and the number of data packets it monitored. We formulate the above two problems as ILP and stochastic programming problems, respectively.

To the best of our knowledge, our work is the first to consider the monitors' activity pattern assumption. Furthermore, this is the first work to apply *stochastic programming* (SP) on random traffic patterns to maximize the amount of collected information. Our contribution can be summarized as follows:

1) For DTP, we formulate the optimal channel assignment as an ILP and we show that its integer optimal solution can be found in cubic time, by proving that its constraints' coefficient matrix is always totally unimodular.
2) For STP, we formulate an SP that breaks the randomness of the traffic into many cases each one of them has a fixed pattern and then we obtain an optimal closed form solution for each of sub-cases.
3) We show by simulation the effect of changing the system parameters on the quality of packet monitoring. Furthermore, we show that our stochastic program outperforms the risk- averse traffic scenario and has a small gap with the perfect information scenario.

The rest of this paper is organized as follows. In Section II, we describe the system model. The problem formulation and solution is depicted in Section III. Section IV presents the simulation results. Finally, Section V concludes the paper.

II. SYSTEM MODEL

Assume that we have a set of legitimate primary users (PUs) having a certain activity pattern over a set of N independent and identically distributed (i.i.d) lossless channels forming the set $\mathcal{C} = \{1, 2..n..N\}$. We also have a set of M mobile users (MUs) $\mathcal{U} = \{1, 2..k..M\}$ communicate over those channels. Some of the MUs tend to violate the rules by communicating without a permit or transmitting with a high power which interferes with the PUs. The violators set is $\mathcal{U}_v = \{1, 2...M_v\}$ and $M_v < M$. To detect the violations, as described in Fig. 1, a set M_d of trustworthy monitors $\mathcal{U}_d = \{1, 2..j..M_d\}$ try to monitor the occupied channels and detect if there are MUs violating the spectrum or not. The monitoring happens at each time slot t, where $t \in \{1, 2...T\}$.

To form good knowledge about violations, the monitors must collect as many packets as possible, which is the main point in this work. The monitors then report their results to a central platform which takes actions against violators such as banning them from accessing the spectrum in the future. Packets aggregation and interpretation, punishments and any platform or PUs' end protocols are out of our scope. As mentioned before, we assume that the monitors are MUs

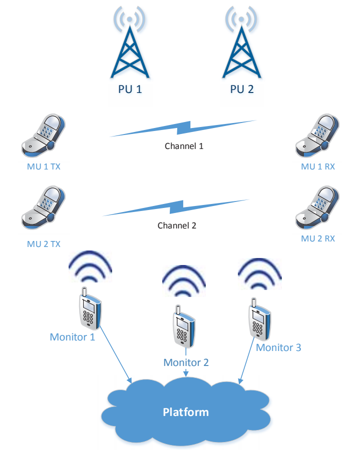

Figure 1: Monitoring System Architecture

which dedicate part of their free time to observe the primary channels. In order to make sure that monitoring won't interfere with the MUs' communication, we construct an availability matrix $A \in R^{M_d \text{x} T}$ which indicates the MUs activity such that $\forall a_{jt} \in A$, $a_{jt} = 0$ if the monitor is not available and $a_{jt} = 1$ if the monitor is available. We assume that the channel condition/gain won't affect the quality of monitoring. Also, we assume that the monitors can deliver the monitoring results to the platform without any loss. In other words, the result that a monitor j finds after monitoring channel n is a true and non-corrupted result.

As we assume that the monitors are real users which the platform rely on, the platform must consider their loss due to monitoring and give them a payment as a compensation. Therefore, the platform should give incentives to the monitors in form of payment p_j, where p_j is more than or equal to the monitoring cost. As said before, the main job of the monitor j is to collect as many packets as it can from the channel n that j is optimally assigned for. We assume that each channel n has a unique traffic pattern that models the traffic communicated on it. We have two cases for the traffic patterns. First, we assume that the traffic pattern $V \in R^{N \text{x} T}$ that characterizes the number of packets at a certain channel at a specific time slot, is deterministic. Second, we assume that the traffic pattern $\widetilde{V} \in R^{N \text{x} T}$ of the packets is an RV and we know its distribution. This assumption is reasonable because although traffic statistics can change over time, there are machine learning techniques to learn and update the traffic distribution online, for example, [7] and [6]. The commonly used notations are summarized in TABLE I.

III. PROBLEM FORMULATION AND SOLUTION

In this section, we formulate and solve our problem in two cases: the deterministic and the stochastic cases.

A. DTP Problem Formulation and Solution

The platform's objective is to maximize the total collected packets and minimize the payments given to the monitors by

Table I: Summary of Frequently Used Symbols

$N/M_d/T$	Number of channels/monitors/time slots
M_d	Number of monitors/detectors
T	Number of time slots
A	Availability matrix
p_j/p	Payment element/vector
$v_n^t/v/V$	Constant traffic pattern element/vector/matrix
$\widetilde{v}_n^t/\widetilde{v}/\widetilde{V}$	Stochastic traffic pattern element/vector/matrix
$z_{jn}^t/z/Z$	Assignment decision element/vector/matrix

optimally assigning M_d monitors to N channels over T time slots ahead without affecting the monitors' communication. DTP problem can be formulated the as follows:

$$DTP : \max_{Z,P} \sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{j=1}^{M_d} v_n^t z_{jn}^t - \beta \sum_{j=1}^{M_d} p_j$$

$$\begin{aligned} \text{s.t.} \quad z_{jn}^t &\in \{0,1\} \quad \forall t,\ \forall j,\ \forall n && (1)\\ \sum_{n=1}^{N} z_{jn}^t &\leq 1 \quad \forall t,\ \forall j && (2)\\ \sum_{j=1}^{M_d} z_{jn}^t &\leq 1 \quad \forall t,\ \forall n && (3)\\ Z_n &\preceq A \quad \forall n && (4)\\ p_j - \sum_{t=1}^{T}(\sum_{n=1}^{N} z_{jn}^t) &\geq 0 \quad \forall j && (5) \end{aligned}$$

where $Z \in R^{M_d \text{x} N \text{x} T}$ is a binary decision variable such that: $z_{jn}^t = 1$ if channel n is monitored at time slot t by monitor j and $z_{jn}^t = 1$ otherwise, β is a weighting factor used by the platform to control the effect of the payment, and v_n^t is the number of packets collected at channel n on time t. In the objective function, we are trying to maximize the difference between the overall number of the collected packets and the payments given to the monitors. In (2), we make sure that each monitor senses one channel at most. In (3), we make sure that each channel is sensed at most once. In (4), we make sure that the channels are only assigned to the available monitors, where Z_n is the matrix of decision elements that corresponds to the same channel n. In (5), we make sure that the payment given to the monitor is at least equal to its monitoring cost.

We can see that DTP is an ILP due to the existence of a binary decision variable. A straight forward solution is obtained by relaxing DTP to an LP and round its optimal value, but this may affect the optimality of the solution. However, if the following conditions are met, we can obtain an optimal integer solution for our problem without the need of relaxation or using exhaustive search, which is NP-Hard and comes with exponential complexity. For any ILP on the form

$$\begin{aligned} &\min_{x} c^\top x \\ \text{s.t. } &Dx \leq b \end{aligned} \qquad (6)$$

where x is an integer decision variable, c, b are vectors, D is the matrix of the constraints' coefficients. There is an integer solution [12] for this problem if D is a totally unimodular (TU) matrix [13]. Meaning that the determinant of any sub-matrix of D can only take values $\in \{-1, 0, 1\}$ and also any element of D must be $-1, 0$ or 1, also b should be a vector of integers. In this case, the optimal solution is integer and could be found on the vertices of the constraints' polyhedron using the *Simplex* method. This solution reduces the complexity from exponential time, like in the exhaustive search, to a cubic time of order $O(M_d^3)$, which is more efficient.

Theorem 1. *DTP has an optimal integer solution.*

Proof. In order to prove the theorem, we should prove that the two TU properties are true for DTP.

1) b is an integer vector, which is true for DTP as all the right-hand side vectors and matrices of DTP constraints, A for example, are integers and have values $\in \{0, 1\}$.
2) The coefficients of the decision variables of DTP form a matrix, this matrix must be TU. This matrix was defined as D, but we call it the coefficients of constraints matrix (CCM). To prove that CCM is TU, we do the following: We partition CCM into small parts, we begin with a small part and prove it is TU then, we add another part and we prove the resultant matrix is TU and so on.

□

Our proof steps depend on the theorems and propositions found in p540, Proposition 2.1 in [14] and p280, 43(V) in [15] about the properties of the TU matrix. Due to the space limitation, we omit the mathematical proof. For more information, visit our technical report [16].

B. *STP Problem Formulation*

In this case, we consider that $\widetilde{V}$ has a finite number of outcomes L and each outcome happens with probability q_l, where $\sum_{l=1}^{L} q_l = 1$. In other words, we know each outcome but we do not know which one will happen. Consequently, choosing a realization by a random guess is a solution with the maximum risk as it results in non-optimal assignment and wrong payments. Therefore, to deal with a problem where a RV exists, we should formulate our problem as a stochastic program. *Stochastic Programming* (SP) [17], [18] is a framework for modelling optimization problems that involve uncertainty or RVs. Whereas deterministic optimization problems are formulated with fixed parameters, real world problems almost invariably include some stochastic parameters. SP takes advantage of the fact that probability distributions governing the data are known or can be estimated. In our work, we resolve the randomness in the stochastic traffic pattern problem (STP) by doing the following: First, we divide the problem into two parts: a fixed part (first stage) and a stochastic part (second stage) using the *recourse formulation* method [19]. We leave the first stage for now, and divide the stochastic second stage into L realization and solve a fixed sub-problem for each realization. Afterwards, we feed the average of second stage sub-problems' solutions to the first stage. And by doing that, we eliminate the stochastic part of the problem. Then, we solve the fixed first stage after taking in consideration all the possible realizations. But first, let's explain the recourse formulation method. A typical stochastic program can be written as follows:

$$\min_{x} \; c^{\top}x + Q(x) \quad \text{s.t. } Dx \leq b \tag{7}$$

where x is the fixed or the first stage objective variable, $c^{\top}x$ is the fixed part of the objective and $Q(x)$ is the stochastic part of the problem and it contains the stochastic objective and the stochastic constraints. $Q(x)$ is given by:

$$Q(x) = E_{\eta}(Q(x, \eta(\omega_l))) \tag{8}$$

where η is a RV takes values $\in \{\omega_1, \omega_2..\omega_l..\omega_L\}$. Each $Q(x, \eta(\omega_l))$ is a fixed problem w.r.t one outcome $\eta(\omega_l)$ and it's given by solving the following optimization problem:

$$Q(x, \eta(\omega_l)) : \min_{y} \; f_i(x, y(\eta(\omega_l))) \quad \text{s.t. } g_i(\eta(\omega_l), y) \leq 0 \tag{9}$$

where $f_i(x, y(\eta(\omega_l)))$ is the objective function and y is any decision variable depends on any stochastic part of the problem. $g_i(\eta(\omega_l), y)$ is the set of all stochastic constraints. The formulation in (7) is called recourse formulation [20] and it's used to solve the stochastic parts separately and then feeds them to the first stage problem as fixed quantities. Using the same concepts, let's first formulate our problem in an extensive-form representation as follows:

$$STP_{ext} : \max_{\mathbf{z},\mathbf{p}} \; \widetilde{v}^T z - \beta 1^T p$$

$$\text{s.t.} \quad z_{jn}^t \in \{0,1\} \quad \forall t,\ \forall j,\ \forall n \tag{10}$$

$$\sum_{n=1}^{N} z_{jn}^t \leq 1 \quad \forall t,\ \forall j \tag{11}$$

$$\sum_{j=1}^{M_d} z_{jn}^t \leq 1 \quad \forall t,\ \forall n \tag{12}$$

$$Z_n \preceq A \quad \forall n \tag{13}$$

$$p_j \geq \alpha \sum_{t=1}^{T}\sum_{n=1}^{N} v_n^t z_{jn}^t + \gamma \sum_{t=1}^{T}(\sum_{n=1}^{N} z_{jn}^t) \quad \forall j \tag{14}$$

where $\widetilde{v}$, z and p are the same as $\widetilde{V}$, Z and P, respectively, but in a vector form. α is a constant represents how much more energy the monitor spends in processing than in being idle. We adopt the value of α from IEEE 802.11 power consumption analysis [21]. Note that we have improved the constraint (5) into (14) to take into consideration the number of packets that monitor j has collected while lower bounding the payment, where γ is a factor represents the monitoring cost per channel per time slot. The new constraint makes the problem not TU, but we will see later that using the recourse formulation will solve this problem as this constraint is related to the stochastic and has no effect on the problem after solving the fixed sub-problems in the second stage. $\widetilde{v}$ can take any value v_l with probability q_l for all $l \in \{1, 2...L\}$. This problem can be transformed to the recourse formulation as follows:

$$STP_{Rec} : \max_{\mathbf{z}} \; Q(z)$$

$$\text{s.t.} \quad z_{jn}^t \in \{0,1\} \quad \forall t,\ \forall j,\ \forall n \tag{15}$$

$$\sum_{n=1}^{N} z_{jn}^t \leq 1 \quad \forall t\ \forall j \tag{16}$$

$$\sum_{j=1}^{M_d} z_{jn}^t \leq 1 \quad \forall t,\ \forall n \tag{17}$$

$$Z_n \preceq A \quad \forall n \tag{18}$$

Note that STP_{Rec} does not have a fixed part in its objective as in (7), as the objective depends on z, p and they are related to the stochastic traffic pattern, but this does not mean we cannot separate the problem into two parts. So that, our first stage fixed problem can be considered as a feasibility problem in which we choose the values of z that are in the feasible region. As in (8), $Q(z)$ is given by:

$$Q(z) = E_{\widetilde{v}}(Q(z, v_l)) \tag{19}$$

and $Q(z, v_l)$ is obtained by solving the following problem:

$$Q(z, v_l) = \min_{\mathbf{p}} \; c^T p - v_l^T z \quad \text{s.t. } Dp \leq b \tag{20}$$

where $c = \beta\mathbf{1}$, $D = -I \in R^{M_d \text{x} M_d}$ and $b \in R^{M_d}$ is a vector where each element $b_j = \sum_{t=1}^{T}(\sum_{n=1}^{N}(\alpha v_n^t + \gamma) z_{jn}^t)\ \forall j$.

The fixed second stage sub-problem for realization l $Q(z, v_l)$ has an optimal closed form solution $p^* = D^{-1}b$ if D is a square matrix and $Dc \preceq 0$ [12]. For our sub-problem, the two conditions are met because $D = -I$ which is absolutely a square matrix and $Dc = -\beta\mathbf{1}^T$ which is always negative. Therefore, generally, $Q(z, v_l)$ is given as follows

$$\begin{aligned} Q(z, v_l) &= \beta\mathbf{1}^T D^{-1} b - v_l^T z \\ &= ((\alpha - 1)v_l^T + \gamma\mathbf{1}^T)z \end{aligned} \tag{21}$$

Hence, $Q(z)$ is given by

$$Q(z) = ((\alpha - 1)\widehat{v}^T + \gamma\mathbf{1}^T)z \tag{22}$$

where $\widehat{v}$ is the expectation of the L realizations. Finally, the STP becomes:

$$STP : \max_{\mathbf{z}}((1-\alpha)\widehat{v}^T - \gamma\mathbf{1}^T)z$$

$$\text{s.t.} \quad z_{jn}^t \in \{0,1\} \quad \forall t,\ \forall j,\ \forall n \tag{23}$$

$$\sum_{n=1}^{N} z_{jn}^t \leq 1 \quad \forall t\ \forall j \tag{24}$$

$$\sum_{j=1}^{M_d} z_{jn}^t \leq 1 \quad \forall t,\ \forall n \tag{25}$$

$$Z_n \preceq A \quad \forall n \tag{26}$$

Table II: Simulation Parameters

T	5	γ, β	1	α	0.2
Slot time	10 ms	packet time	1 ms	L	1000
Iterations	1000	λ	5	A_p	70%

which is a simple LP with a TU constraint matrix as well, but we won't proof that the matrix of Constraints' coefficients of STP is TU because it is the same as the DTP, but with one constraint less, which would not change the TU property.

The time complexity of the STP scheme is on the of order of $O(2^{\lambda})$ (exponential time) as we solve a sub-problem for all the possible realizations. Note that the time complexity of each sub-problem is cubic of order $O(M_d^3)$, but the overall solution is dominated by the exponential time .

IV. Numerical Results

In this section, we conduct extensive simulations to study the behavior of our schemes: DTP and STP. First, we study the behavior of DTP using data obtained from real applications. Second, for STP, we study the effect of changing synthetic data on monitor selection. Then, we compare STP with two existing schemes.

In the simulation setting, we generate the availability matrix A using a normal RV with probability pr of having ones more than zeros. In other words, we say that we have availability probability $A_{pr} = 70\%$, when we have a probability of having monitors = 0.7. We construct the realizations set from the highest probability realizations belong to the set of all possible realizations, because we can't simulate all the realizations as their number goes to ∞. The lth realization probability q_l is calculated by $q_l = \prod_{i=1}^{NT} \rho_i$, where ρ_i is the probability of having exactly b packets under Poisson distribution. The rest of the parameters are given in TABLE II.

The procedure of the simulations for STP works as follows: First, we generate the traffic pattern on each channel in each time slot from a Poisson distribution with mean $\lambda = 5$ and we generate the the availability as mentioned above. Second, we select the L realizations having the highest probability. Third, we feed this information to the problem, which we write in the form of "intlinprog" program in MATLAB. Forth, we solve the monitor-to-channel problem and use that to calculate the overall number of collected packets (Pck). Finally, we change the system parameters like $M_d, N, \beta ...etc$ to measure their effect on Pck.

For DTP, we use real data traces collected from 802.11g WLAN network, [7]. In their setting, they consider five different types of trace data (FTP, BT, Web Browsing, Skype Voice and Skype Video). As they did, we construct five channels directly from the mentioned data traces, and we construct another five from the mixes between them. Therefore, we have up to ten channels to monitor. Fig. 2 uses this data to draw Pck against M_d. We can see that the Pck as M_d increase until M_d becomes bigger enough to collect all the packets, where the curve saturates.

In Fig. 3, we plot Pck of STP versus M_d for different values of N. We can see that the more the number of channels the more the collected packets. Also, we can see that the number of collected packets saturates after certain M_d as happens in DTP. We can learn here that having a huge M_d is not always efficient because the more the monitors the more the power and the payments spent. Therefore, we can achieve the same performance with less number of monitors if they achieve the monitoring goal. We can see almost the same insights in Fig. 4 where the payment increases as the number of monitored channels increases. Furthermore, a curve that plots Pck against A_{pr} would have been a valid curve, but due to the space limitation, it was omitted.

In Fig. 5, we plot Pck against the weighting factor β. In the beginning, as β increases the problem solution remains the same, until a point where β starts to make the platform favors minimizing the payment over maximizing the packet collecting. At this point, Pck starts to fall.

In Fig. 6, we draw the number of collected packets against N. Generally, we can see that as the number of channels increases, the number of collected packets increases. However, the curve is divided into two trends. A fast trend, which happens when M is relatively bigger than N. In this case, the monitors collect as many packets as possible as their availability is enough and a slow trend when N begins to grow relative to M, and the monitors becomes insufficient to sense all the channels which slows collecting packets.

In Fig. 7, we compare STP with two other schemes from stochastic programming paradigm. First, the worst-case scenario scheme where the monitors are assigned to the channels with the lowest number of packets. This scheme is Risk-averse, meaning that it favors being more conservative than to risk by taking any arbitrary traffic pattern. Second, the perfect information scenario, where scenario that will happen is assumed to be known for sure and we solve STP for this scenario, then we take the weighted average of all scenarios. Our scheme proves to be better than the worst-case scenario which means that STP achieves more wealth for the same amount of risk. On the other hand, STP is outperformed by the perfect scheme as it assumes the complete knowledge of the traffic pattern. The gap between the former scheme and STP is called the expected value of perfect information (EVPI). EVPI is the price that one would be willing to pay in order to gain access to perfect information, which is low in our problem.

V. Conclusion

Crowdsourcing is a powerful tool that enables the platform to hire a huge number of service providers, such as mobile users, to complete certain tasks that the platform defines which saves the resources of the platform and provides more accurate information. In this paper, we adopt the crowdsourcing notion to maximize the collected information from the channels which we suspect that violations exist upon them. We optimally assign the monitors to channels in T time slots ahead where we consider two cases. First, the deterministic case, where the traffic pattern is known. Second, the stochastic

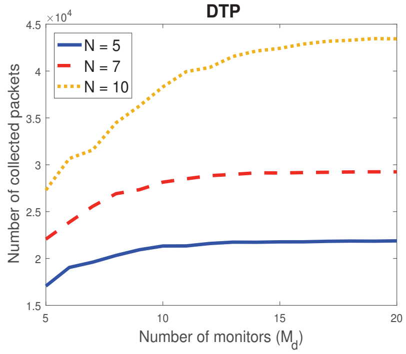

Figure 2: Pck vs. M_d for different N

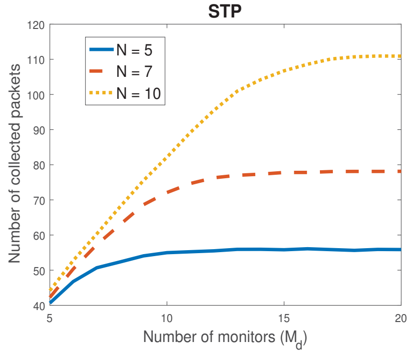

Figure 3: Pck vs. M_d for different N

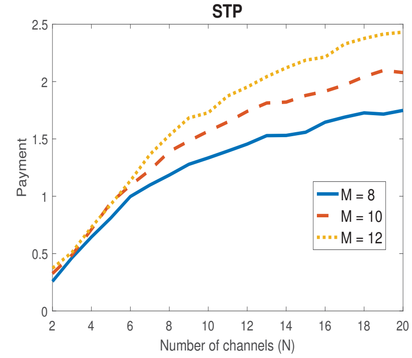

Figure 4: p vs. N for different M_d

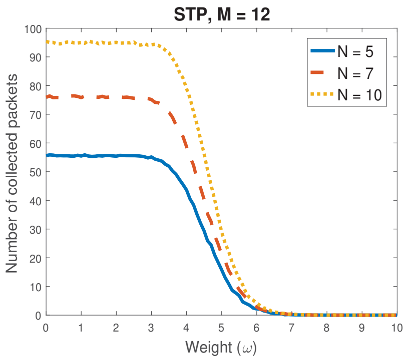

Figure 5: Pck vs. ω for different N

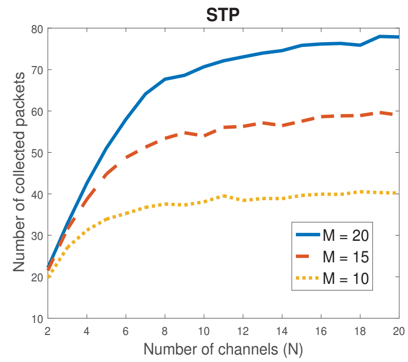

Figure 6: Pck vs. N for different M_d

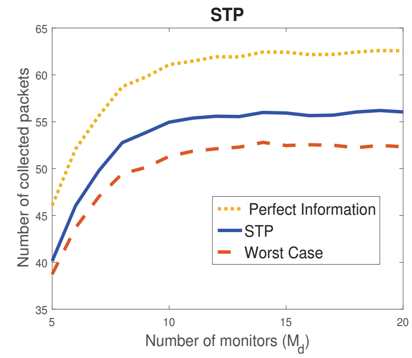

Figure 7: Pck vs. M_d for different SPs

case, where the traffic pattern is a RV with known distribution. The simulations show that our scheme outperforms the Risk-averse scenario. Furthermore, we have shown that EVPI is small in our case which strengthens our model. For the future work, we intend to construct an approximation algorithm to decrease the complexity of STP, add the switching cost as a parameter in the setting and perform traffic pattern estimation instead of assuming the knowledge of the traffic distribution.

References

[1] F. C. Commission *et al.*, "Spectrum policy task force," *rep. ET Docket*, vol. 2, p. 135, 2002.
[2] J. Mitola III and G. Q. Maguire Jr, "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, no. 4, pp. 13–18, 1999.
[3] J.-M. Park, J. H. Reed, A. Beex, T. C. Clancy, V. Kumar, and B. Bahrak, "Security and enforcement in spectrum sharing," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 270–281, 2014.
[4] A. Dutta and M. Chiang, ""see something, say something" crowdsourced enforcement of spectrum policies," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 67–80, 2016.
[5] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 173–184.
[6] S. Chen, K. Zeng, and P. Mohapatra, "Efficient data capturing for network forensics in cognitive radio networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 6, pp. 1988–2000, 2014.
[7] Q. Yan, M. Li, F. Chen, T. Jiang, W. Lou, Y. T. Hou, and C.-T. Lu, "Spec-monitor: Toward efficient passive traffic monitoring for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5893–5905, 2014.
[8] D.-H. Shin and S. Bagchi, "Optimal monitoring in multi-channel multi-radio wireless mesh networks," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2009, pp. 229–238.
[9] A. Chhetri, H. Nguyen, G. Scalosub, and R. Zheng, "On quality of monitoring for multi-channel wireless infrastructure networks," in *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2010, pp. 111–120.
[10] J. Xu, J. Xiang, and D. Yang, "Incentive mechanisms for time window dependent tasks in mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6353–6364, 2015.
[11] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Keep your promise: Mechanism design against free-riding and false-reporting in crowdsourcing," *IEEE Internet of Things Journal*, vol. 2, no. 6, pp. 562–572, 2015.
[12] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[13] P. Camion, "Characterization of totally unimodular matrices," *Proceedings of the American Mathematical Society*, vol. 16, no. 5, pp. 1068–1073, 1965.
[14] G. L. Nemhauser and L. A. Wolsey, "Integer programming and combinatorial optimization," *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, vol. 20, pp. 8–12, 1988.
[15] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.
[16] A. Salama, M. Li, and D. Yang, "Optimal Crowdsourced Channel Monitoring in Cognitive Radio Networks (Technical report)," http://wiser.arizona.edu/papers/globecom17-techrep.pdf, 2017.
[17] P. Kali and S. W. Wallace, *Stochastic programming*. Springer, 1994.
[18] J. R. Birge and F. Louveaux, *Introduction to stochastic programming*. Springer Science & Business Media, 2011.
[19] A. Shapiro, D. Dentcheva *et al.*, *Lectures on stochastic programming: modeling and theory*. SIAM, 2014, vol. 16.
[20] S. Ahmed and N. V. Sahinidis, "Robust process planning under uncertainty," *Industrial & Engineering Chemistry Research*, vol. 37, no. 5, pp. 1883–1892, 1998.
[21] D. d. O. Cunha, L. H. M. Costa, and O. C. M. Duarte, "Analyzing the energy consumption of ieee 802.11 ad hoc networks," in *Mobile and Wireless Communication Networks*. Springer, 2005, pp. 473–484.