Data Driven Algorithms to Tune Physical Layer Parameters of an Underwater Communication Link

Satish Shankar
ARL, Tropical Marine Science Institute,
National University of Singapore,
Singapore 119227.
mailshanx@gmail.com

Mandar Chitre
ARL, Tropical Marine Science Institute,
National University of Singapore,
Singapore 119227.
mandar@arl.nus.edu.sg

Melani Jayasuria
Department of Electrical
and Computer Engineering,
National University of Singapore,
Singapore 117576.
melani@nus.edu.sg

***Abstract*— The design of a communication system is closely linked to assumptions about the nature of the channel. A vast majority of the components of a modern communication system are implemented in software, affording us the ability to fine tune their parameters during operation. The objective for tuning the parameters could be to optimize data rates, protect against errors, minimize power, and so on. If the physics of the channel is completely known, it is possible to determine the values of these parameters for a given objective. However in practice, it is quite difficult to know the state of the channel completely. The parameters usually interact with each other, so tuning them in isolation is often not possible. We present a data driven approach for tuning the physical layer parameters of a communication link to optimize data rates, assuming the channel remains static over the course of a file transfer. Our approach does not need any knowledge of the physics of the channel. We illustrate the application of our approach in the context of an underwater communication link.**

I. Introduction

A. Problem description

We consider the problem of transferring a finite sized file in the minimum possible time using an underwater acoustic link. We assume that the channel remains static over the course of the file transfer. The file will be transferred over multiple packet bursts. At every transmission, we may choose to tune the modem parameter values. Since we do not have any prior information about the average data rate resulting from any parameter choice, we need to choose between exploring new parameter values versus exploiting the parameter values that have yielded the best results so far. The objective is to devise a strategy to balance this exploration and exploitation so that the file transfer time is minimum.

B. Choice of modem parameters

As an illustrative example of modem parameters that can be tuned, consider the modem implementation in [1]. The OFDM based modem had three choices for modulation: 2-differential phase shift keying (DPSK), 4-DPSK and 8-DPSK. There were also two forward error correction (FEC) encoders. Each of the FEC encoders had $\frac{1}{4}$, $\frac{1}{3}$, and $\frac{1}{2}$ rate codes, thus resulting in a total of 27 possible configurations, out of which 10 were tested and the corresponding bit error rates (BER) were recorded. For a packet size of 8000 bits, the BER from the paper can be converted into packet error rates (PER) assuming that the errors are independent. The average data rate is then simply the product of the uncoded data rate, code rate and the PER. These are tabulated in Table I. It is clear from the table that the 2-DSPK modulation scheme used with a 1/4 rate code gives the best average data rate. However, this choice can only be inferred from the BER information that is obtained by sampling the channel.

TABLE I
Average data rate for various link and coding schemes

Modulation	Code Rate	BER	Avg. Data Rate (bps)
2-DPSK	1/2	1.2×10^{-2}	0
2-DPSK	1/3	3.2×10^{-4}	137
2-DPSK	1/4	$< 10^{-4}$	449
4-DPSK	1/3	1.6×10^{-2}	0
4-DPSK	1/4	1.5×10^{-3}	0.012
4-DPSK	1/6	$< 10^{-4}$	399
8-DPSK	1/4	7.7×10^{-2}	0
8-DPSK	1/8	2.9×10^{-2}	0
8-DPSK	1/12	4.3×10^{-4}	10
8-DPSK	1/16	$< 10^{-4}$	84

Besides the choice of modulation scheme and error control coding, examples of other tunable parameters may include the number of sub-carriers, prefix length, suffix length, peak-to-average-power-ratio (PAPR) parameters, and so on. More examples of underwater communication systems can be found in [2] and [3].

C. Explore or exploit?

We have two choices at every transmission:

1) Try out new link parameter values, thus exploring the parameter search space.
2) Exploit the parameter values with the highest observed data rate.

Exploration contributes to the knowledge base and improves channel knowledge. Improved channel knowledge enables us to choose the parameter set with the highest average data rate and therefore minimize the time taken to transfer the file. Every exploration is associated with a cost: time is lost and the result of the transmission may or may not be successful. The state of the existing knowledge base and the size of the file remaining to be transferred are key factors in deciding between exploring and exploiting. For example, if the size of the file is very large or if we know nothing about the channel, it is probably better to spend some time exploring the search space before settling on the best parameter values. On the other hand; if we have a high confidence on the accuracy of our channel estimate, or if the size of the remaining file is very small, it is probably better to exploit the best known parameter value to completion. Thus an ideal decision policy would be one that consistently makes an appropriate choice in light of expected rewards, existing knowledge, and remaining file size.

D. Solution strategies

A brute force strategy would systematically try all parameter values, record the resulting data rate, and select the one with the best recorded data rate to transfer the rest of the file. This approach entails a comprehensive exploration of the parameter search space, and makes an educated choice based on the results of the exploration. An alternative to brute force searching would be a first hit strategy. The algorithm randomly selects parameter values till it finds one with an average data rate greater than a certain threshold, and uses it to transfer the rest of the file. Yet another strategy would be to simply randomly select parameter values at every transmission. We develop some strategies that try to balance the approaches of brute force, first hit and random selection and compare their performance.

Note that though we assume a static channel, in reality the channel will only be quasi-static, i.e. it stays static for a short period of time and then changes state. The results of a brute force search may not be useful if the channel changes state before the search terminates. The brute force or first hit strategies explore at the beginning, and then exploit – hence they are unable to track any changes in the channel during the exploitation phase. We develop an adaptive strategy that has a capability to track changes in a slowly changing channel.

II. Problem Formulation

A. Definitions

1) Choice of link schemes, FEC codes, and payload sizes: Let the size of the file to be transferred be F. We define the n-tuple consisting of all tunable parameters of a communication system excluding FEC coding and packet size as a *link scheme*. Let the communication modem have m link schemes available, each of them resulting in an uncoded link data rate of d_i bits per second, $i \in \mathbb{Z}^+, i \leq m$. Let $c \in \mathcal{C}$ denote all available code rates and let $l \in \mathcal{L}$ denote all available packet lengths. Any code can be combined with any other packet length subject to the constraint $l/c \leq M$, where M denotes the maximum length of a coded packet supported by the modem. We define a set $\mathcal{A} = \mathcal{C} \times \mathcal{L}$ containing all feasible combinations of codes and packet lengths. The set $\mathcal{A}$ is indexed by $j \in \mathbb{Z}^+$, $j \leq n$. Let the packet header size be h bits. Let γ_j be a discount factor to account for the effect of the code rate and the overhead of the packet header:

$$\gamma_j = \frac{c_j l_j}{l_j + h} \tag{1}$$

γ_j is ordered such that $\gamma_j \leq \gamma_{j+1}$. We term the elements $a \in A$ as *coding schemes*, and arrange them in decreasing order of the discount factor γ_j.

2) Channel representation: The PER for all possible link-coding scheme combinations describe the channel. Thus, the channel is represented by the $m \times n$ channel PER matrix E_{ij}^{ch}. The algorithm has no knowledge of E_{ij}^{ch} but tries to estimate it based on data.

3) Sequences and bursts: It is possible to change parameter values used for every single packet transmission. However in practice, the process of tuning often involves a handshaking based protocol. The transmitter will need to initiate the process by specifying the parameter values and wait for an acknowledgement. Let the time taken by the protocol to change parameter values be equal to t^{p}. Due to the time overhead t^{p} imposed by the protocol, for efficiency, we may wish to transfer the file in packet bursts. Larger bursts also allow us to estimate the PER more accurately by measuring the fraction of transmitted packets that are received successfully. Let b^{tx} be the size of a packet burst. The file transfer will involve the transmission of a sequence of bursts. Let $S = \{(S_u^{\text{I}}, S_u^{\text{J}}) \forall u\}$ denote such a sequence where $i = S_u^{\text{I}}$ is the link scheme and $j = S_u^{\text{J}}$ is the coding scheme selected at burst u. For brevity, we define the notation $i(u)$ to mean S_u^{I} and $j(u)$ to mean S_u^{J}. The amount of data transferred successfully f_u^{xfer} by burst u is given by

$$f_u^{\text{xfer}} = b_u^{\text{rx}} c_{j(u)} l_{j(u)} \tag{2}$$

where b_u^{rx} is the number of successfully received packets in burst u. b_u^{rx} is a random variable which follows a binomial distribution with success parameter $1 - E_{ij}^{\text{ch}}$ and trial length b^{tx}:

$$b_u^{\text{rx}} \sim B(b^{\text{tx}}, 1 - E_{i(u),j(u)}^{\text{ch}}) \tag{3}$$

The time t_u taken to transfer f_u^{xfer} amount of data is given by

$$t_u = \frac{(l_{j(u)} + h) b^{\text{tx}}}{d_{i(u)}} + t^{\text{p}} \tag{4}$$

4) Problem statement: Let the time taken to transfer the entire file corresponding to sequence S be T^{S}. We then have:

$$T^{\text{S}} = \sum_u t_u \tag{5}$$

$$\sum_u f_u^{\text{xfer}} = F \tag{6}$$

Given a file of finite size F, the problem is to compute sequence S to minimize the expected value of T^S.

III. Solution Strategies

A. Brute force search

In this strategy, all link-coding scheme combinations are systematically tried. At every transmission, a data burst of size b^{tx} packets is transmitted. The brute force average data rate matrix D^{BF} is calculated as

$$D_{i(u),j(u)}^{\text{BF}} = d_{i(u)} \gamma_{j(u)} \frac{b_u^{\text{rx}}}{b^{\text{tx}}} \quad \forall\, u \tag{7}$$

After all link schemes have been tried, we can reasonably conclude that the link scheme with the best observed average data rate is most suitable for transmitting the rest of the file, and use the link-coding scheme combination corresponding to the maximum of D_{ij}^{BF} for transmitting the remaining file. However, if mn is large and/or the file size is small, then the time taken to explore the parameter space mn may be the primary contributor to the total file transfer time, resulting in a low average data rate.

B. *Random selection*

In this strategy, no attempt is made to explore the search space. Link scheme i is randomly chosen in the interval $[1, m]$. Similarly, coding scheme j is also randomly selected from the interval $[1, n]$. The procedure continues to the end of the file transfer.

C. *First hit strategy*

In this strategy, the algorithm randomly selects link-code scheme combinations till it comes across one with an data rate greater than a threshold data rate or until it exhausts all mn link-code schemes, after which the best combination is used for the rest of the file transfer.

D. *ϵ-greedy strategy*

Brute force, random selection and first hit strategies resolve the explore vs. exploit dilemma by taking extreme measures of either exploring the entire search space comprehensively, or undertaking very less, if any, exploration at all. An ϵ-greedy strategy provides a way to balance exploration and exploitation. The extent to which new link-code scheme combinations are explored is governed by the parameter $\epsilon \in [0, 1]$, which is set at initialization. New link-code scheme combinations are selected randomly with a probability of ϵ and the ϵ-greedy average data rate matrix D_{ij}^{ϵ} is updated as follows:

$$D_{i(u),j(u)}^{\epsilon} = d_{i(u)}\gamma_{j(u)}\frac{b_u^{\mathrm{rx}}}{b^{\mathrm{tx}}} \quad \forall\, u \tag{8}$$

At all other instances, the link-code scheme combination corresponding to $\arg\max_{ij} D_{ij}^{\epsilon}$ is used.

E. *Bayesian inference algorithm*

1) Transitory points: Let the BER after FEC decoding be B. Assuming the errors are independent, the resultant PER is given by

$$P = 1 - (1 - B)^{l_j + h} \tag{9}$$

Fig. 1 plots P vs. B for $l_j = 8000$ and $h = 2000$. Because of the exponential form of (9), coupled with the fact that the number of bits in a packet is typically very large, P is very sensitive to B. Increasing B ever so slightly saturates P to unity. Changing the packet size changes the region of B values over which P rises to 1.

In general, decreasing code rates increases robustness and decreases P at the cost of data rate for a given decoding complexity. Since the increase in P with increasing B is very sharp, when the coding schemes $a \in \mathcal{A}$ are arranged

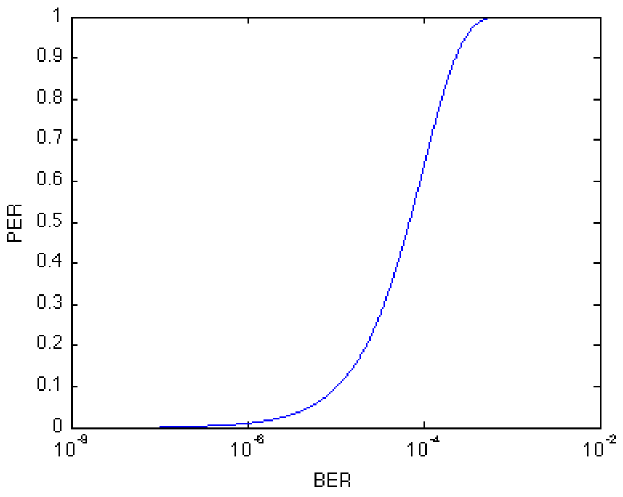

Fig. 1. PER vs. BER for $l_j = 8000$ bits and $h = 2000$ bits

in decreasing order of data rate for a given link scheme, P goes from 1 to 0 rapidly. We assume that the coding schemes are spaced out far enough apart, such that not more than one coding scheme may have $0 < P < 1$. This imposes a special structure to the channel matrix. We can then define a *transition point* w_i for link scheme i such that:

$$E_{ij}^{\mathrm{ch}} = \begin{cases} 1 & j < w_i \\ 0 & j > w_i \\ \alpha & j = w_i,\ 0 < \alpha \leq 1 \end{cases} \tag{10}$$

We note that for a given link scheme, the coding scheme used must be either $j = w_i$ or $j = w_{i+1}$ for maximum data rate. For $j < w_i$, the $P = 1$, and no packets are received. For $j > w_i$, the $P = 0$, but the data rate is sub-optimal.

The existence of such transitory points is central to the proposed algorithm. We apply a Bayesian inference technique to locate the transitions for each link scheme, and hence the name. We also assume that consecutive coding schemes are spaced out far enough apart such that not more than 1 coding scheme may lie at the transition for any given link scheme. However, as evident from (10), the transition itself may not necessarily coincide with any coding scheme.

2) The algorithm: Let Δ_{ij}^{u} be the probability that the coding scheme j is the transitory coding scheme for link scheme i, based on the accumulated knowledge before the transmission of burst u. Initially, since we have no knowledge of the location of the transitory point, we set $\Delta_{ij}^{1} = 1/n\ \forall\, i, j$. We derive E_{ij}^{u}, an estimate of E_{ij}^{ch}, from Δ_{ij}^{u}:

$$E_{ij}^{u} = \begin{cases} \sum_{\hat{j}=j+1}^{n} \Delta_{i\hat{j}}^{u} & u = 1 \\ \Delta_{ij}^{u} E_{ij}^{u-1} + \sum_{\hat{j}=j+1}^{n} \Delta_{i\hat{j}}^{u} & u > 1 \end{cases} \tag{11}$$

We select link-coding scheme $i(u), j(u)$ with the maximum estimated data rate for transmission during burst u such that:

$$(i(u), j(u)) = \arg\max_{(i,j)} d_i \gamma_j E_{ij}^{u} \tag{12}$$

A burst consists of a series of b^{tx} packets with the associated acknowledgements. We use the number of successfully received packets in the burst b_u^{rx} to update our estimate of the channel. We have no information on the link scheme that was not selected for this burst. Hence:

$$\Delta_{ij}^{u+1} = \Delta_{ij}^{u} \quad \forall\, i \neq i(u) \tag{13}$$

For the link scheme $i(u)$ selected for transmission, we perform a Bayesian update of the probability of transition based on the measured PER $b_u^{\text{rx}}/b^{\text{tx}}$ of the current burst. We consider 3 different cases based on the PER. If $b_u^{\text{rx}}/b^{\text{tx}} = 0$ then:

$$\Delta_{i(u),j}^{u+1} = \frac{1}{\nu}\frac{1}{n-j(u)}\Delta_{i(u),j}^{u} \qquad \forall\, j > j(u) \tag{14}$$

$$\Delta_{i(u),j}^{u+1} = 0 \qquad \forall\, j \leq j(u) \tag{15}$$

where $\nu = \sum_{j=1}^{n} \Delta_{i(u),j}^{u+1}$ is the normalizing constant. If $b_u^{\text{rx}}/b^{\text{tx}} = 1$ then:

$$\Delta_{i(u),j}^{u+1} = \frac{1}{\nu}\frac{1}{j(u)-1}\Delta_{i(u),j}^{u} \qquad \forall\, j < j(u) \tag{16}$$

$$\Delta_{i(u),j}^{u+1} = 0 \qquad \forall\, j \geq j(u) \tag{17}$$

For all other values of $b_u^{\text{rx}}/b^{\text{tx}}$ we use:

$$\Delta_{i(u),j(u)}^{u+1} = 1 \tag{18}$$

$$\Delta_{i(u),j}^{u+1} = 0 \qquad \forall\, j \neq j(u) \tag{19}$$

With the updated Δ_{ij}^{u+1} values, we repeat the process starting with (11) for the next burst $u+1$.

IV. Simulation Results

Uncoded data rates d we selected randomly from the interval between 2000 to 3000 bits per second. $\mathcal{C}$ = {1/20, 1/3, 1/2, 2/3, 3/4} and $\mathcal{L}$ = {40, 100, 500, 1000, 3000, 5000, 8000} were used to generate $\mathcal{A}$. Setting $M = 8000$ resulted in $\mathcal{A}$ having a total of 33 coding schemes. Channel representations E_{ij}^{ch} were generated with $n = 33$ coding schemes and varying number of link schemes, such that link schemes with higher uncoded data rates had low robustness. For any given link scheme i, this requirement was incorporated by setting the location of the transition point from a sigmoid function to which we add Gaussian noise:

$$w_i = \frac{32}{1 + e^{-.05(d_i - 2500)}} + 1 + X \tag{20}$$

X is a random variable that has a normal distribution with zero mean and standard deviation set to 4. Burst size b^{tx} was set to 100 and the protocol overhead time t^{p} was set to 3 seconds. Packet header size h is set to 1000 bits. Also, for every E_{ij}^{ch} matrix, 15 link schemes were selected at random and their transition points were set to $w_i = 1$, in order to simulate the existence of a fixed number of "good" link schemes that become increasingly rare as the search space increases. File transfers using brute force, random selection, ϵ-greedy, first hit strategy and the Bayesian inference algorithm were simulated and the results were averaged over 40 independent trials. For the ϵ-greedy strategy, ϵ is set to 0.1, since this value yielded the best performance in our simulations. Also, the threshold data rate for the first hit strategy is set at 200 bits per second.

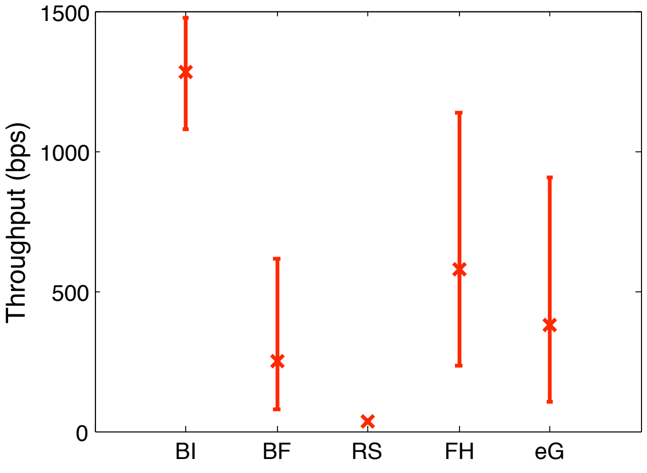

Fig. 2. Comparative throughput for a 5 Mb file transfer with $m = 100$ and $n = 33$. The throughput is computed by dividing the file size by the file transfer time for Bayesian Inference (BI), brute force (BF), random selection (RS), first hit (FH) and ϵ-Greedy (eG) algorithms.

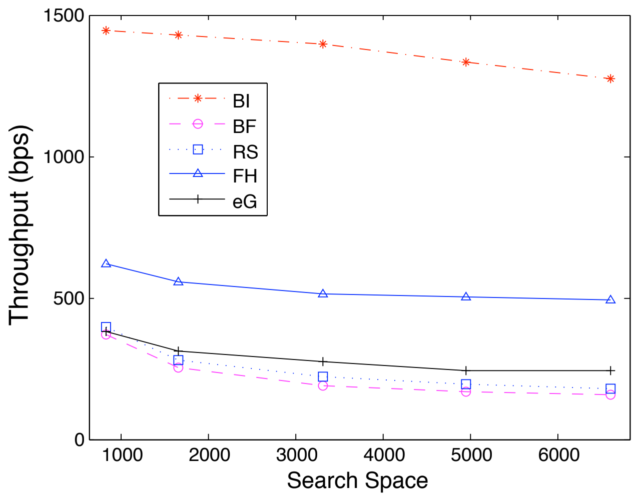

Fig. 3. Variation of throughput with search space size mn for a 5 Mb file, for $n = 33$. The throughput is computed by dividing the file size by the file transfer time for Bayesian Inference (BI), brute force (BF), random selection (RS), first hit (FH) and ϵ-Greedy (eG) algorithms.

Fig. 2 shows the mean throughput with 95% confidence intervals for a typical simulation instance with $F = 5$ Mb over a 100×33 channel. Fig. 2 shows that the Bayesian inference algorithm has the highest mean throughput, followed by first hit, ϵ-greedy, brute force, and random selection.

Fig. 3 shows the variation of throughput with search space for a 5 Mb file. The throughput decreases as the search space increases since the "good" link schemes become increasingly harder to locate. The Bayesian inference algorithm consistently performs better for all the search spaces under consideration.

Fig. 4 shows the variation of throughput with burst size for a 5 Mb file over a 100×33 channel. In general, the curves follow a bell shape with a maxima. This follows from the fact

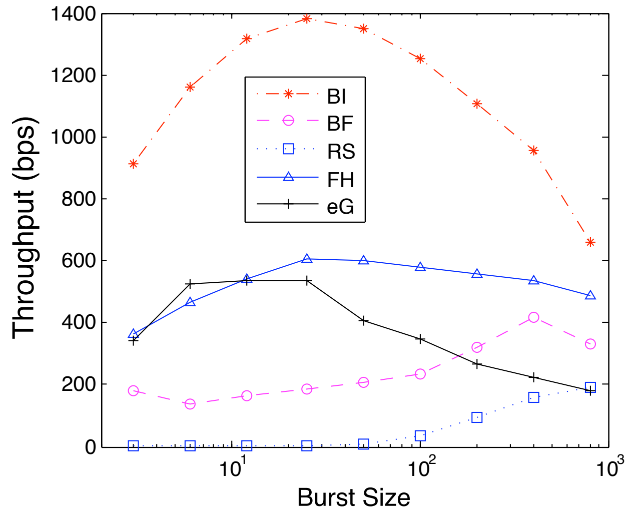

Fig. 4. Variation of throughput with burst size b^{tx}. The throughput is computed by dividing the file size by the file transfer time for Bayesian Inference (BI), brute force (BF), random selection (RS), first hit (FH) and ϵ-Greedy (eG) algorithms.

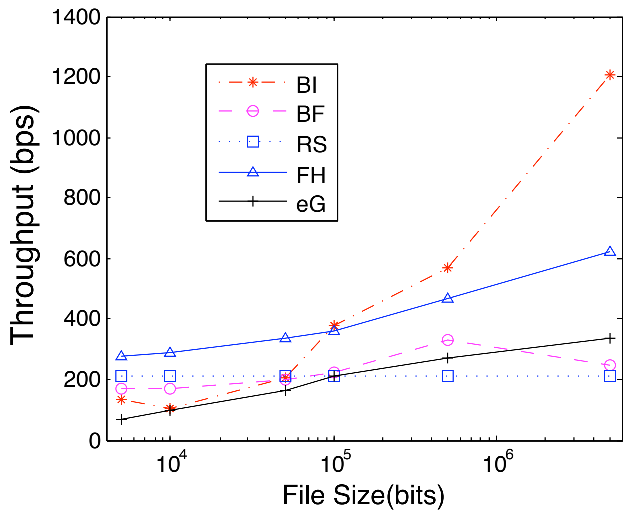

Fig. 5. Variation of throughput with file size. The throughput is computed by dividing the file size by the file transfer time for Bayesian Inference (BI), brute force (BF), random selection (RS), first hit (FH) and ϵ-Greedy (eG) algorithms.

that transmitting very small burst sizes is not efficient due to the protocol overhead. Transmitting very large bursts is not efficient either since a lot of time may be lost if a large data burst fails to get successfully received.

Fig. 5 shows the variation of throughput with file size for a 100×33 channel. It is evident that for relatively small file sizes, the throughput curves are close to each other, confirming the intuition that if a file is very small, random selection is as good a strategy as any other. In fact, for very small file sizes, it is indeed inefficient to explore and therefore the Bayesian inference algorithm performs poorer than the random selection or the first hit algorithms. As the file size increases, the performance gains from the Bayesian inference are apparent.

V. Conclusion

We have illustrated various possible data driven algorithms to tune physical layer parameters. We investigated the effect of varying search space, burst size and file size for each algorithm. Simulation results show that a Bayesian inference based algorithm performs the best for most cases, except when file sizes are very small. The algorithms are independent of the physical layer implementation, and can easily be implemented as link tuners for any modem with tunable parameters.

Acknowledgment

This research work was partly funded by the National University of Singapore and the Ministry of Education of Singapore, AcRF Tier 1 funding, under Grant No. R-263-000-521-133.

References

[1] M. Chitre, Ong, S.-H., and J. Potter "Performance of coded OFDM in very shallow water channels and snapping shrimp noise," in Proceedings of IEEE/MTS OCEANS'05, Washington DC, 2005.

[2] Zhu Wei-Qing ,Wang Chang-Hong, Pan Feng, Zhu Min, Wang Rui, Zhang Xiang-Jun,Dai Yong-Mei, "Underwater acoustic communication system of AUV," in Proceedings of IEEE/MTS OCEANS'98, France, 1998.

[3] Shuxiang Guo and Zixin Zhao, "Design of a QPSK-CDMA acoustic communication system for multiple underwater vehicles," in Proceedings of ICMA 2009, China, 2009.