THE UNIVERSITY OF ARIZONA
College of Engineering

Dept. of Electrical and Computer Engr.

Improving Performance Robustness of a Cognitive Radio Engine

Hamed Asadi, Haris Volos, Michael Marefat, and Tamal Bose

Dept. of Electrical and Computer Engr.
The University of Arizona
Tucson, AZ 85721-0104
hasadi, hvolos, marefat, tbose@email.arizona.edu

THE UNIVERSITY OF ARIZONA

Agenda

- Introduction
- Problem Formulation
  - Multi-Armed Bandit
  - Exploration vs Exploitation
- Robust Training Algorithm
  - Stationary Environment
    - Results
  - Non-Stationary Environment
    - Forgetfulness factor
    - Robust Training Algorithm
    - Results
- Robust Training in Meta Level
- Summary & Conclusions

Agenda

- **Introduction**
- Problem Formulation
  - Multi-Armed Bandit
  - Exploration vs Exploitation
- Robust Training Algorithm
  - Stationary Environment
    - Results
  - Non-Stationary Environment
    - Forgetfulness factor
    - Robust Training Algorithm
    - Results
- Robust Training in Meta Level
- Summary & Conclusions

Introduction

- Cognitive Radio Engines
- Link Adaptation
- Providing Predictable Performance
- Challenges

Multi-Armed Bandit

μ_1 μ_2 μ_3

Bandit “arms”

(unknown reward)

- Pull arms sequentially so as to maximize the total expected reward
- There are K number of slot machines (possible communication methods)
- R_y is a random variable that represents the returned reward by using each communication method

Problem Formulation

- Belief states $R_y(n)$

$$\pi(n) = [\pi_1(n), \pi_2(n), \dots, \pi_y(n)]^T$$

$$\left(\bar{\mu}_y(n), \bar{\sigma}_y^2(n), n'\right) \text{ , } \left(\overline{\text{PSR\_}\mu}_y(n), \overline{\text{PSR\_}\sigma}_y^2(n), m'\right)$$

- In the MAB problem we are looking for a policy that maximizes the expected return $V(s)$:

$$V(s) = E_\pi \sum_{n=1}^{N} \gamma^n R^n$$

Where N is the maximum number of operations, E_y is the expectation operator over the belief state π, and γ is a discount factor $0 < \gamma < 1$

Exploration vs Exploitation

- A central concern of studies of ***adaptive processes*** is the relation between the exploration of ***new possibilities*** and the exploitation of ***old certainties***.
  - ***Exploration*** includes things captured by terms such as search, variation, risk taking, experimentation, play, flexibility, discovery, innovation.
  - ***Exploitation*** includes such things as refinement, choice, production, efficiency, selection, implementation, execution.
- The crucial tradeoff the CE faces at each trial is:
  1. **"exploitation"**
  2. **"exploration"**

Robust Training

- Training Packets

- The PSR's CI of a communication method is estimated by approximating Beta dist. using a normal dist. With a mean $\hat{\theta} = \frac{\alpha}{m\prime}$ and variance $\sigma^2 = \frac{\hat{\theta}(1-\hat{\theta})}{m\prime}$, where $m' = \alpha + \beta$

$$\hat{\theta}_y - z\alpha/_2\, \sigma < \theta_y < \hat{\theta}_y + z\alpha/_2\, \sigma$$

- Estimating upper and lower bounds of each communication method's throughput by using belief state of each method for $n' < 30\ and\ n' > 30$

$$\bar{\mu}_y(n) - t_{\delta/2,n'-1} \frac{\hat{\sigma}_y}{\sqrt{n'}} < R_y < \bar{\mu}_y(n) + t_{\delta/2,n'-1} \frac{\hat{\sigma}_y}{\sqrt{n'}}$$

$$\bar{\mu}_y(n) - t_{\delta/2} \frac{\hat{\sigma}_y}{\sqrt{n'}} < R_y < \bar{\mu}_y(n) + z_{\delta/2} \frac{\hat{\sigma}_y}{\sqrt{n'}}$$

Robust Training

- Classifying all communication methods to:
  1. Offsetting methods
  2. Ineligible methods
  3. Training methods

- Offsetting methods

$$\min_{Throughput} < R_{ly}(n) \quad \text{AND} \quad \min_{PSR} < \theta_{ly}(n)$$

- Ineligible methods

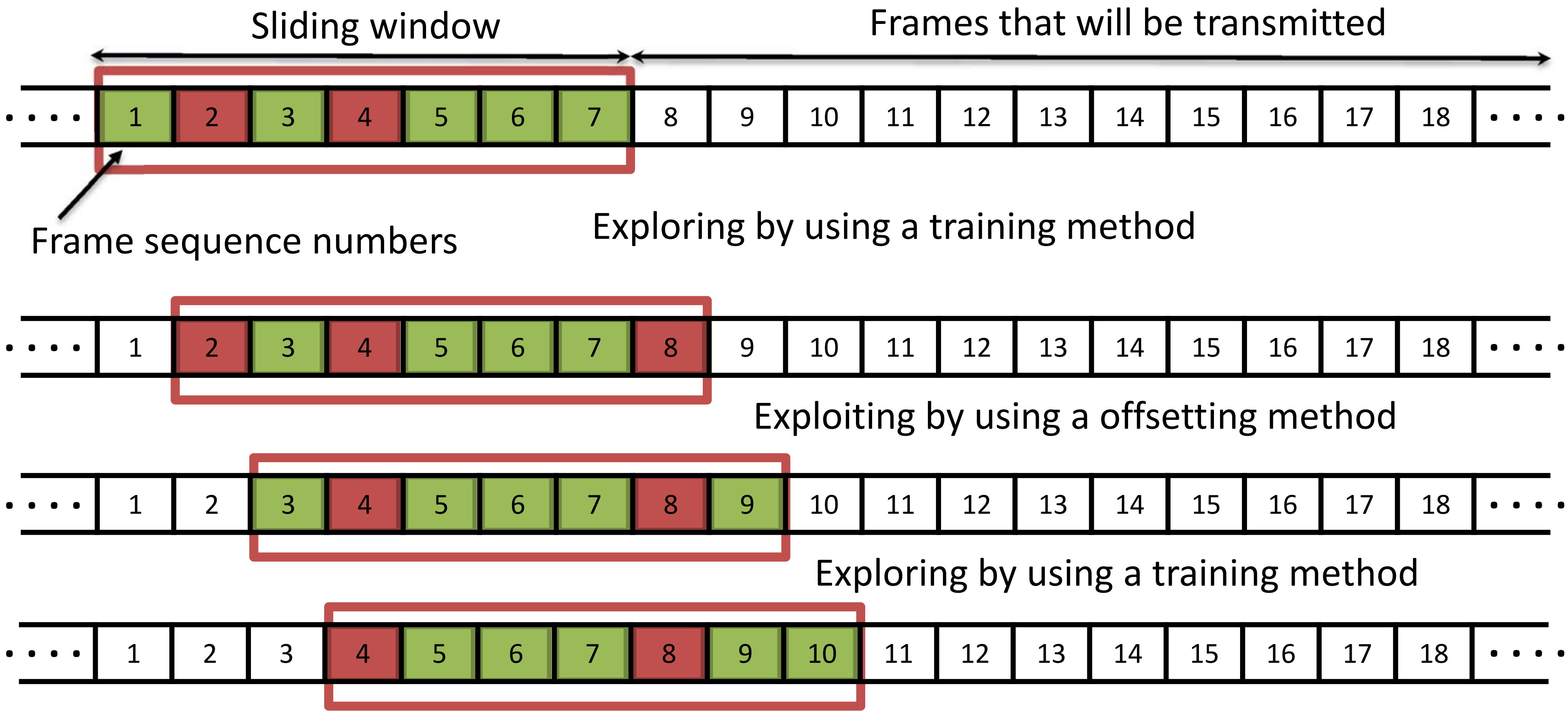
$$R_{uy}(n) < \min_{\text{Throughput}} \quad \text{OR} \quad \theta_{uy}(n) < \min_{\text{PSR}}$$

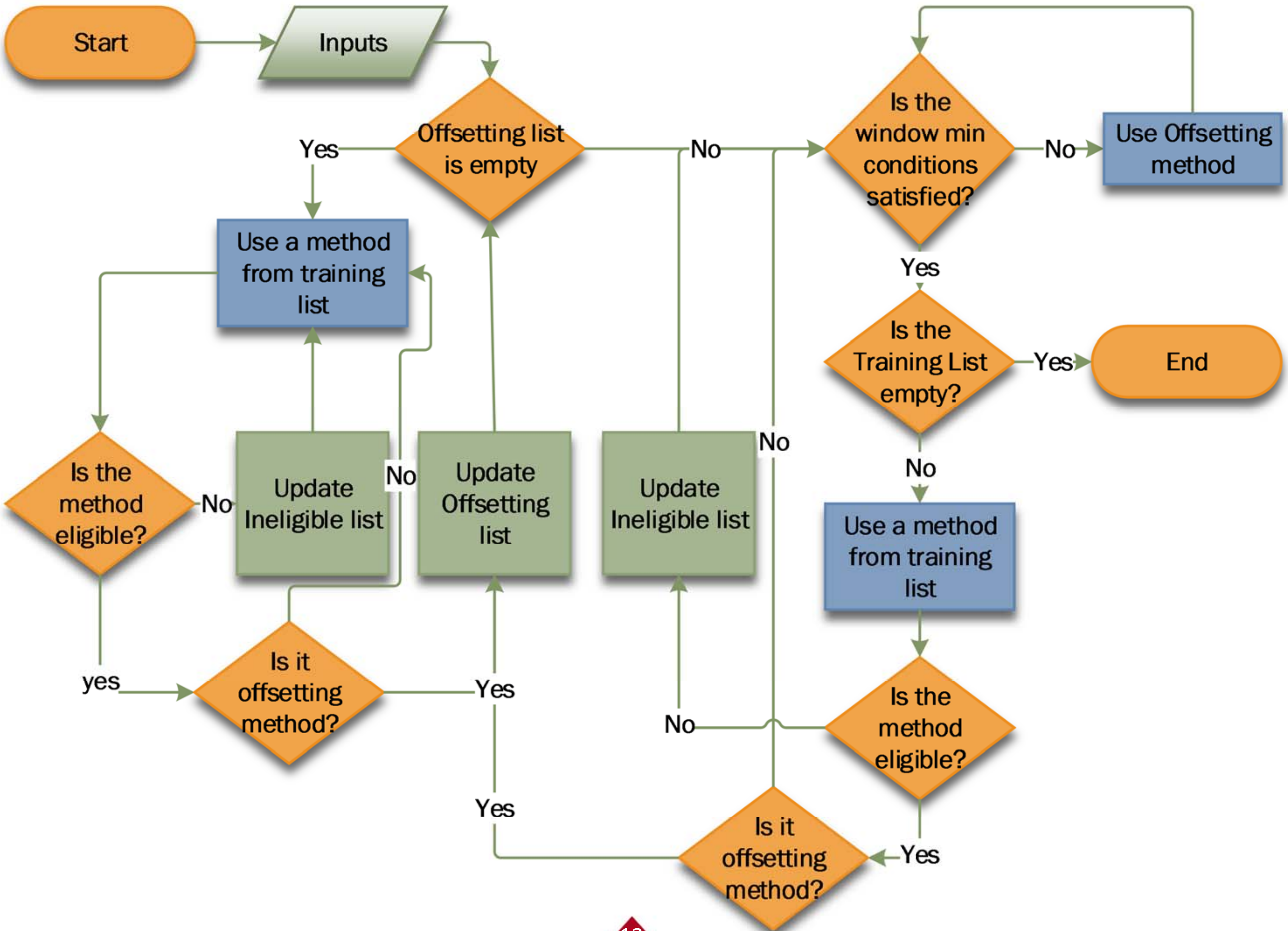
- Training methods

$$R_{ly}(n) < \min_{\text{Throughput}} < R_{uy}(n) \quad \text{AND} \quad \theta_{ly}(n) < \min_{\text{PSR}} < \theta_{uy}(n)$$

Robust Training

(Minimum required PSR) $\theta_{min} = 0.5$

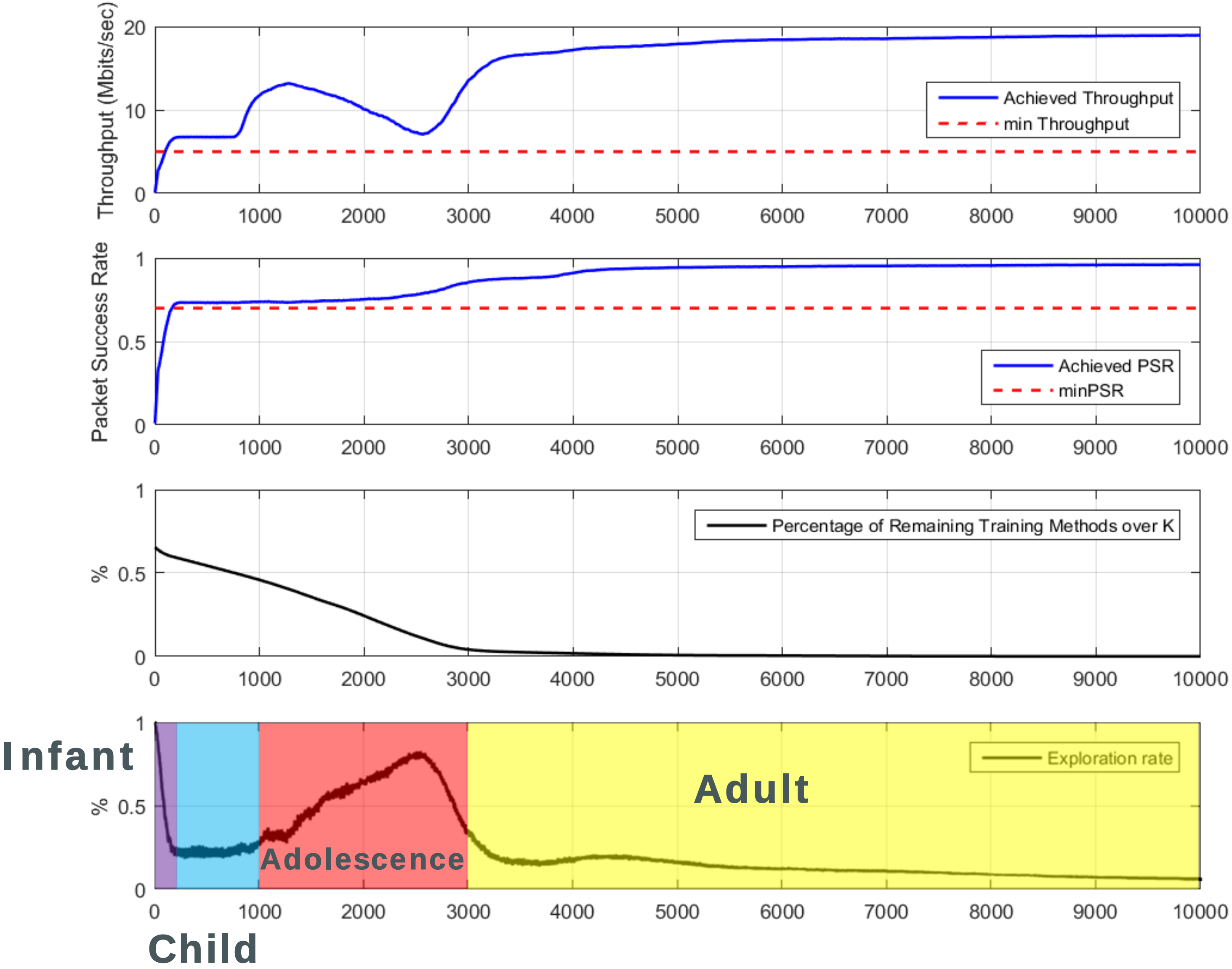



Start
Inputs
Offsetting list is empty
Yes
No
Is the window min conditions satisfied?
No
Use Offsetting method
Yes
Is the Training List empty?
Yes
End
No
Use a method from training list
Is the method eligible?
Yes
No
Is it offsetting method?
No
Yes
Use a method from training list
Is the method eligible?
No
yes
Is it offsetting method?
Yes
No
Update Ineligible list
Update Offsetting list
Update Ineligible list
No

Experimental Results

- Each communication method is defined by a combination of modulation type, coding and antenna techniques:
  - Modulations: QPSK, 8PSK, 16, 32, 64, 128, and 256 QAM
  - Error correction rates: 1, 7/8, 3/4, 2/3, 1/2, 1/4, 1/6, and 1/8
  - Antenna techniques: VBLAST, STBC, and MRC
  - We consider an SNR in the range of 0-50 and the log of the Eigen spread in the range of 0-12 by step size of 0.5
  - There are 5 channels available with different SNR and bandwidth (either 1.25 or 2.5 MHZ)
  - $\theta_{min} = 0.65$ and $R_{min} = 5(Mbits/sec)$

Experimental Results



- Introduction
- Problem Formulation
  - Multi-Armed Bandit
  - Exploration vs Exploitation
- Robust Training Algorithm
  - Stationary Environment
    - Results
  - **Non-Stationary Environment**
    - **Forgetfulness factor**
    - **Robust Training Algorithm**
    - **Results**
- Robust Training in Meta Level
- Summary & Conclusions

Non-Stationary Condition

- Estimated channel conditions change rapidly
- RoTA will not provide the expected performance
- The obtained knowledge by CE about the performance of communication methods will no longer be valid
- RoTA needs a many iterations to transfer its information to the new channel conditions
- This happens because the obtained knowledge is the average of n trials, therefore to overcome the previous knowledge, the RoTA needs to try $m >> n$ trials to wipe out the outdated knowledge

Forgetfulness Factor

- To decrease the weight of old information we suggest to use exponentially-weighted average method

- To estimate the mean and variance of communication methods' performances we use following equations

$$\bar{\mu}_y(n) = \bar{\mu}_y(n-1) + \lambda[R_y(n) - \bar{\mu}_y(n-1)]$$

$$\hat{\sigma}_y^2(n) = (1-\lambda)(\hat{\sigma}_y^2(n-1) + \lambda\left(R_y(n) - \bar{\mu}_y(n-1)\right)^2)$$

Where $0 < \lambda < 1$, can be either constant, or a function of time, $\lambda = \lambda_y(t) = \frac{1}{n}$, as n reaches ∞, by the law of large numbers $\bar{\mu}_y(n)$ converges to the actual mean.

- In the case of constant λ

$$\bar{\mu}_y(n) = (1-\lambda)\bar{\mu}_y(0) + \sum_{i=1}^{n} \lambda(1-\lambda)^{n-i} R_y(n)$$

Since the λ parameter decays the old observations, we call it ***forgetfulness factor***

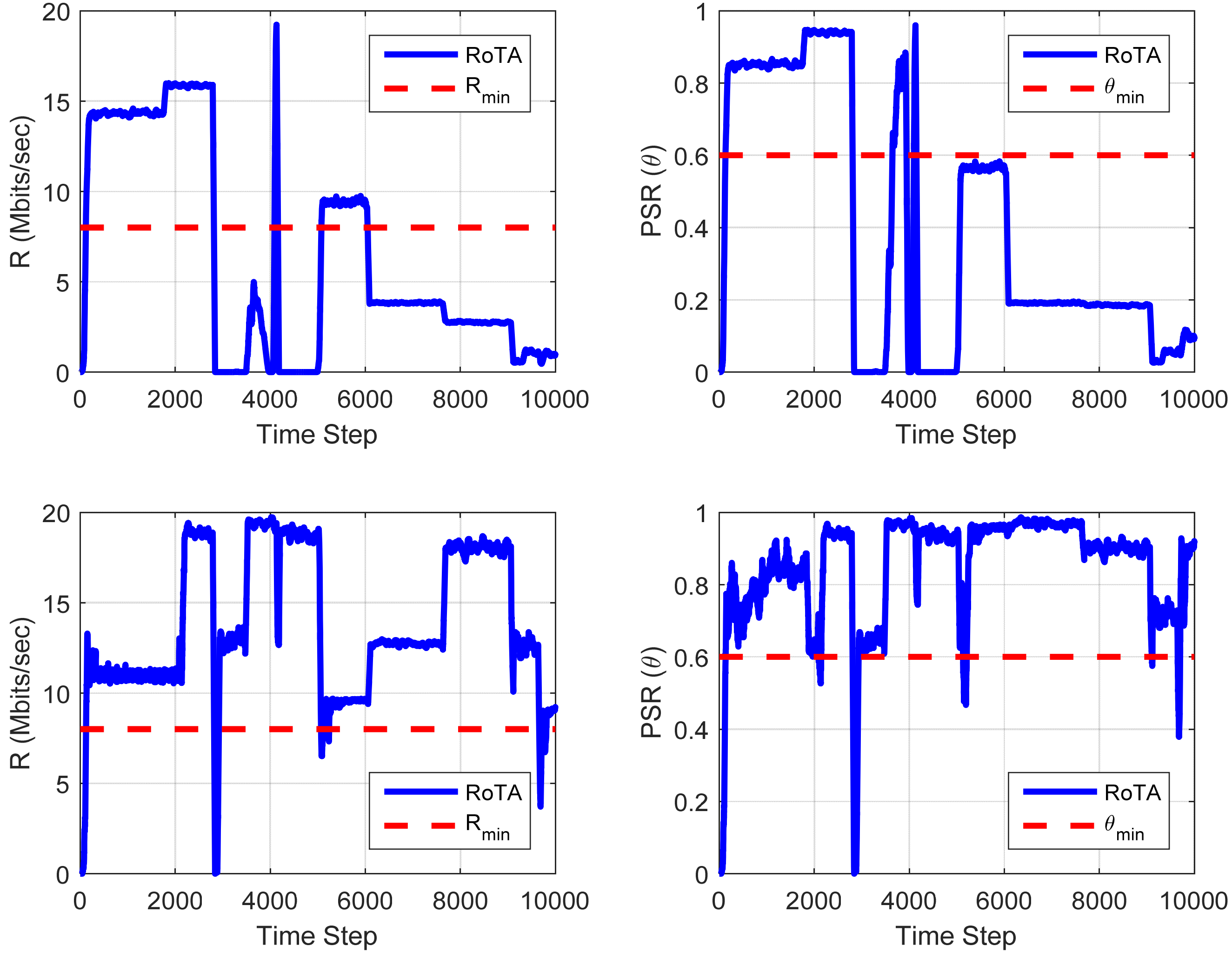
Experimental Results

Time Step	1758
	2790
	3472
	4133
	5029
	6039
	7632
	9066
	9638

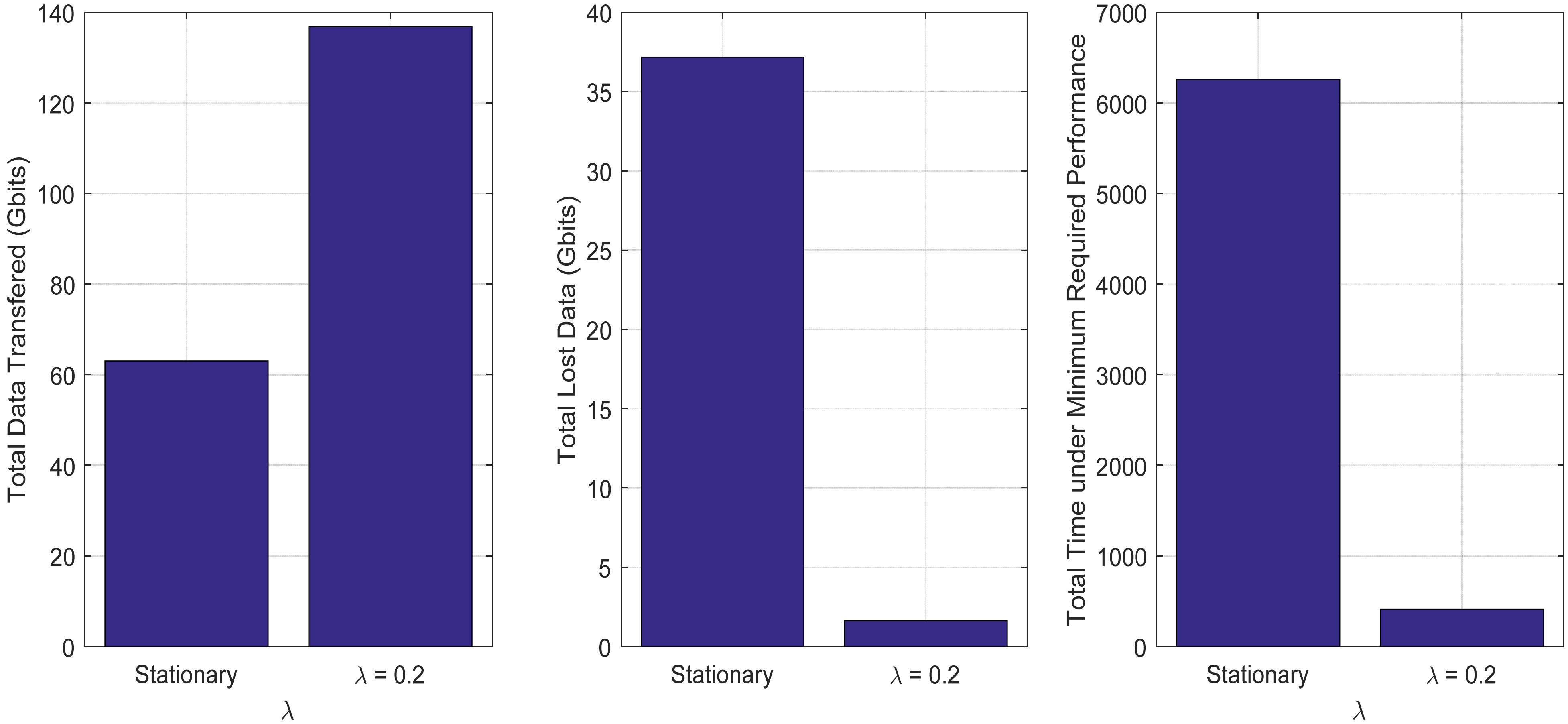
$\theta_{min} = 0.6$

$R_{min} = 8(Mbits/sec)$

$\lambda = 0.2$

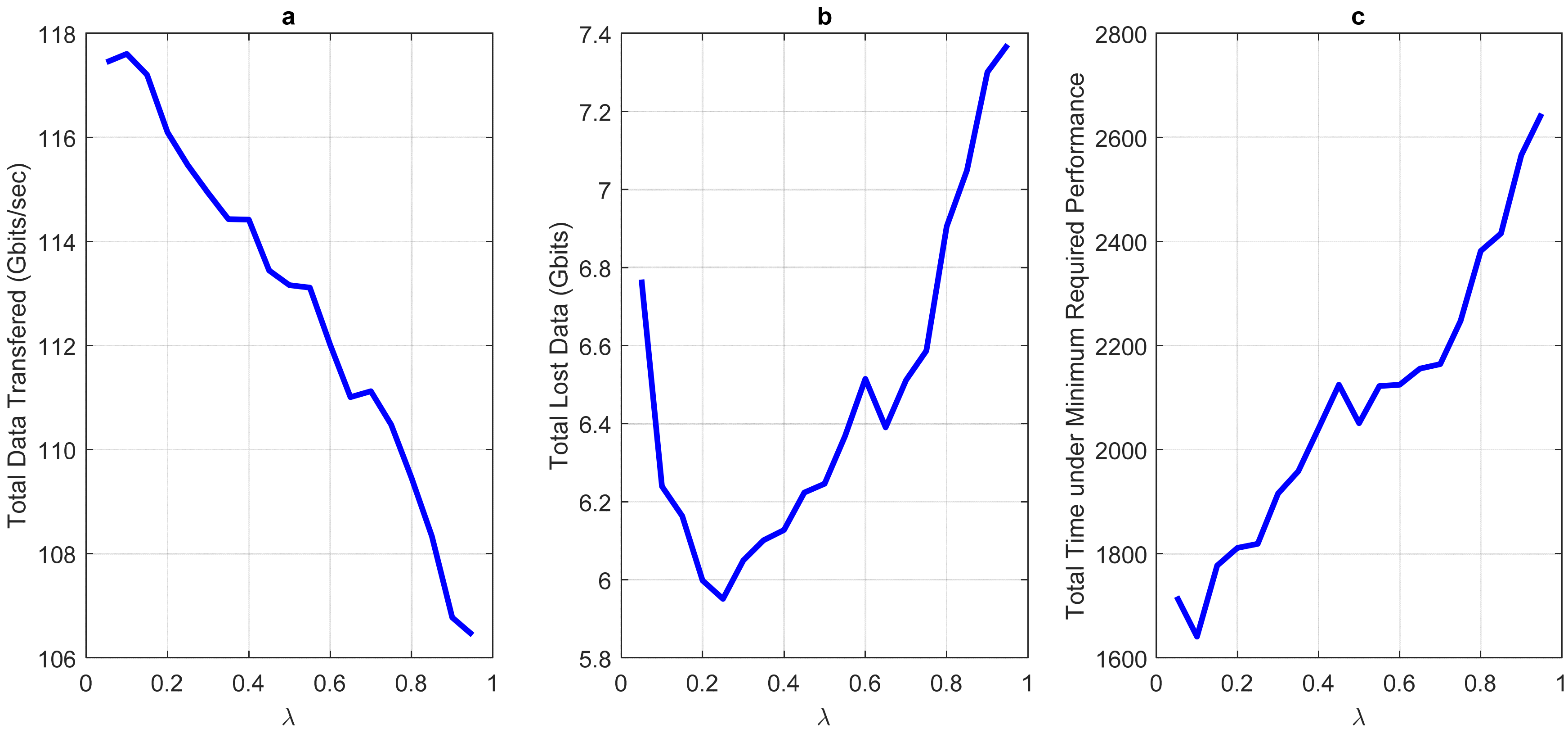


Experimental Results



Experimental Results

- Comparing the effect of different values of λ

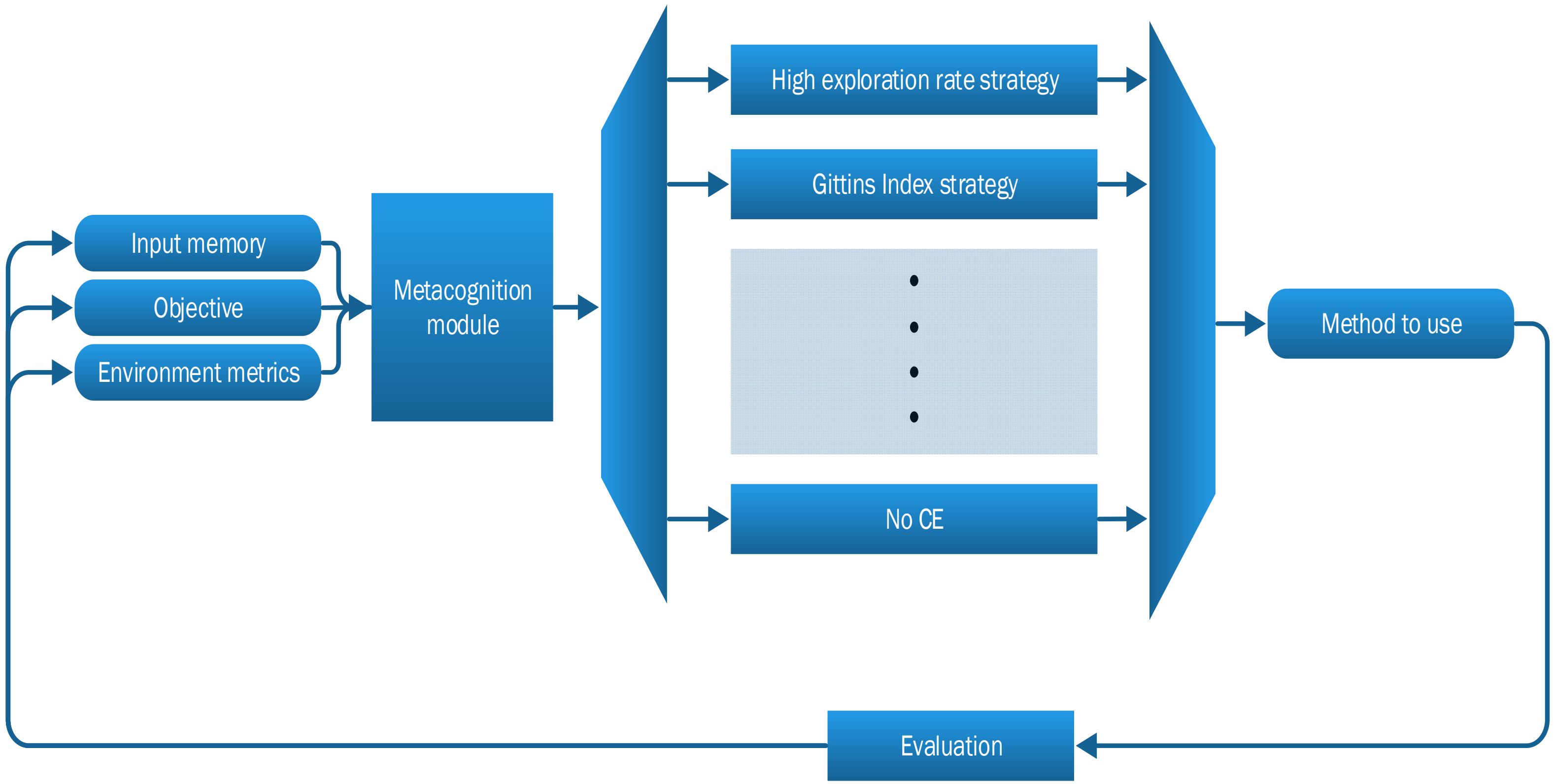


- Introduction
- Problem Formulation
  - Multi-Armed Bandit
  - Exploration vs Exploitation
- Robust Training Algorithm
  - Stationary Environment
    - Results
  - Non-Stationary Environment
    - Forgetfulness factor
    - Robust Training Algorithm
    - Results
- **Robust Training in Meta Level**
- Summary & Conclusions

Robust Training in Meta Level

- Governing individual CE algorithms to operate in various learning stages

- Controlling computation cost of individual CEs



- Introduction
- Problem Formulation
  - Multi-Armed Bandit
  - Exploration vs Exploitation
- Robust Training Algorithm
  - Stationary Environment
    - Results
  - Non-Stationary Environment
    - Forgetfulness factor
    - Robust Training Algorithm
    - Results
- Robust Training in Meta Level
- **Summary & Conclusions**

Summary & Conclusions

- Robust Training Algorithm (RoTA) to guarantee the minimum performance
- Forgetfulness factor to tune the RoTA for various non-stationary environments
- Metacognitive Engine framework to control the complexity of learning algorithms and minimum level of performance

Thank You
Q&A