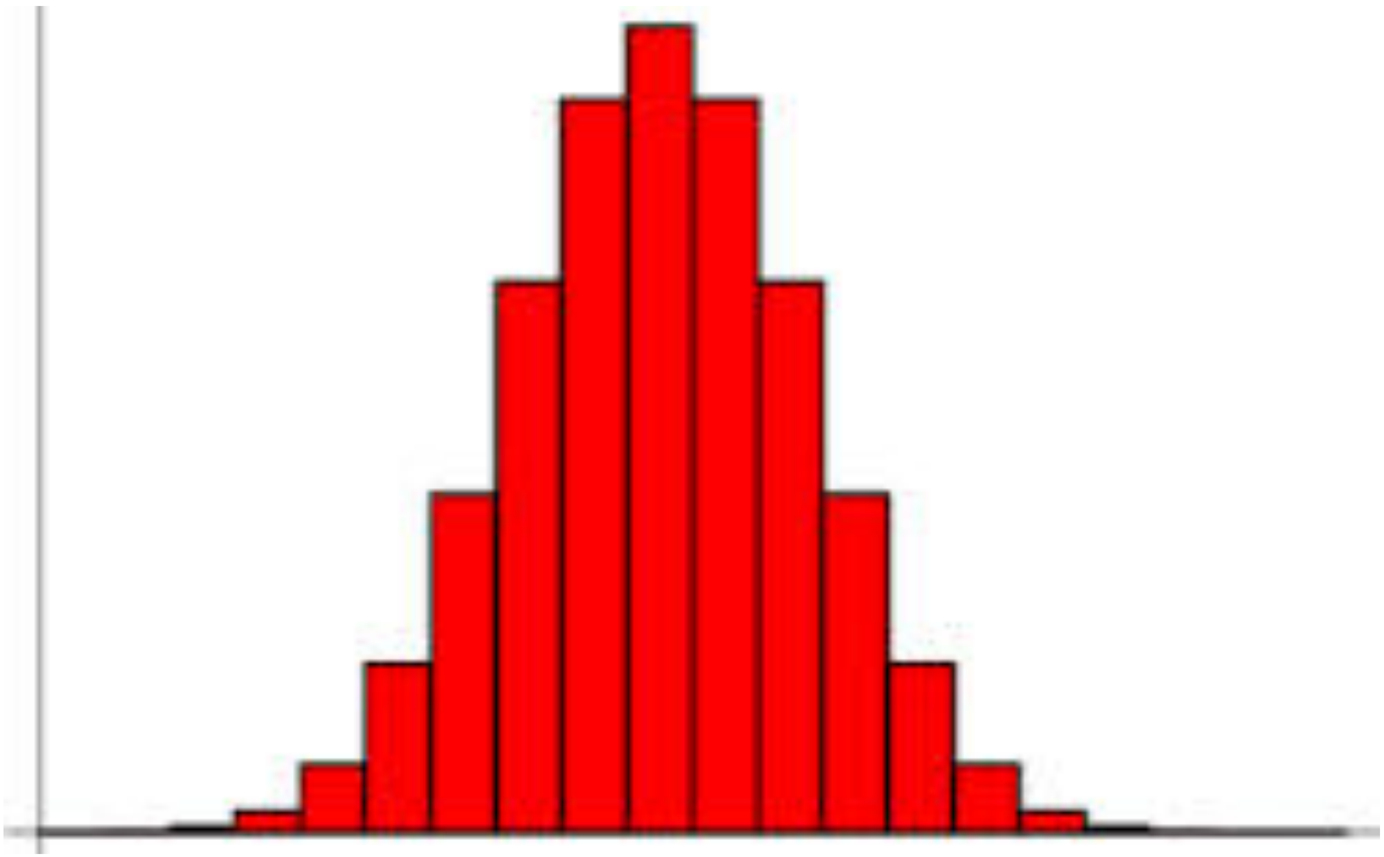
Interpolating Between Stochastic and Worst-case Optimization

R. Ravi

ravi@cmu.edu

Risk – calculable gamble

Handling input risk: Average case analysis



Uncertainty – incalculable unknown

'Symbol of Uncertainty', John Gilbert

Handling input uncertainty: Worst-case competitive analysis

Common complaints

- Competitive analysis is too pessimistic
- Stochastic analysis need too detailed information

Possible Responses

- ***Relax pessimism:*** Add restrictions to worst-case input scenarios to reduce impact
- ***Temper optimism:*** Make stochastic analysis robust to changes in input distribution
- ***"Have it all":*** Build algorithms that achieve the 'best of both worlds' and work for both kinds of inputs
- ***Interpolate Model AND Performance:*** Build hybrid input models interpolating stochastic and adversarial inputs to derive new algorithms that deteriorate smoothly in performance

Outline

- Motivation
- Related work* (incomplete and representative)
- Attempt at Specific Model

Relax Pessimism: Restrict classes of inputs

- Better algorithms for bounded tree-width inputs

- Better approximation algorithms for bounded genus graphs

Relax Pessimism: Smoothed Analysis

- [Spielman-Teng 2001] Uses distributional disturbance over given worst-case input, to smooth out (expected) performance
- Resulting running times are polynomial in input and perturbation size

- Goal: Explain "unreasonable effectiveness" of popular algorithms (like Simplex for LP)

Temper optimism: Stochastic Programming Variants

$$\min_{x \in X} \{c(x) + \mathbb{E}[f_x(\omega) q(x, \omega)]\}$$

$$\mathbb{Q}_x(\omega) \equiv f_x(\omega)\mathbb{P}(\omega)$$

$$\mathcal{Q} = \left\{ f \in L^2(\Omega, \mathscr{B}, \mathbb{P}) : 0 \le f(\omega) \le \frac{1}{1-\alpha} \text{ for all } \omega \in \Omega, \ \langle f, 1 \rangle = 1 \right\}$$

Stochastic Combinatorial Optimization with controllable risk aversion level
So, Zhang & Ye, APPROX '06, Math of OR

Temper Optimism: Distributionally Robust Optimization

Add ambiguity to risk

- Newsvendor problem with only mean and variance, not distribution [Scarf 1958]
- Distributionally Robust Stochastic Optimization [Zackova, Dupacova, Bertsimas+, Sim+, Ye+,…]

Temper Optimism: Correlation Robustness

Correlation Robust Stochastic Optimization

Shipra Agrawal* Yichuan Ding† Amin Saberi ‡ Yinyu Ye §

Abstract

We consider a robust model proposed by Scarf, 1958, for stochastic optimization when only the marginal probabilities of (binary) random variables are given, and the correlation between the random variables is

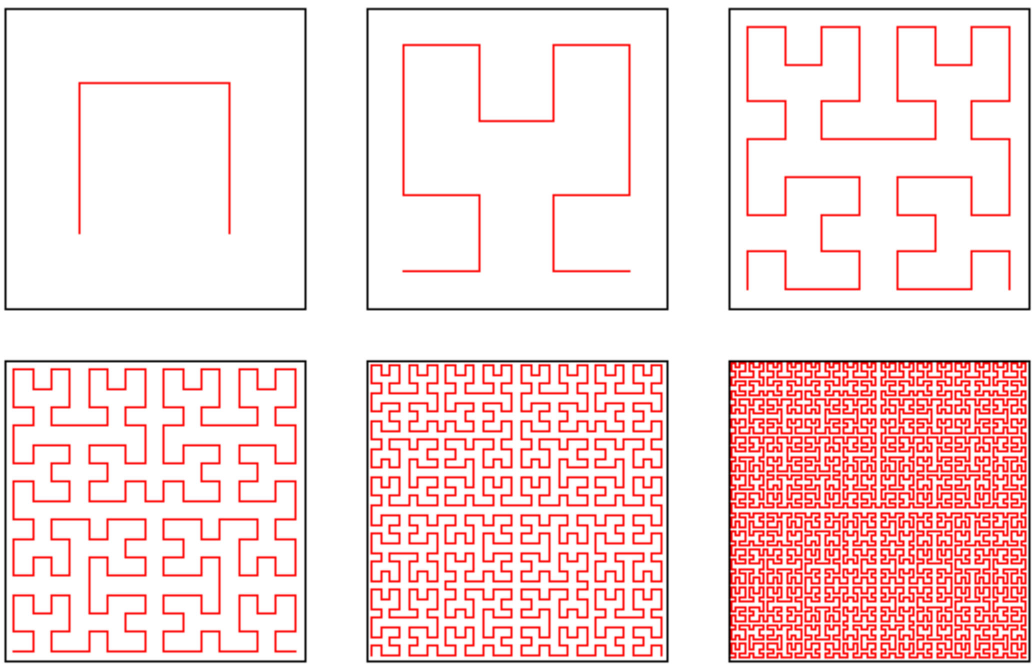
1 Introduction

Stochastic optimization models decision making under uncertain or unknown problem data. We consider stochastic optimization problems in which the uncertain variable is the "demand" set. For example, in stochastic network design problems, the random

Correlation gap: Loss in performance by ignoring correlations and assuming independence

Have it all:
Universal Approximations

One single solution whose induced solution is good for any subset input [Bartholdi-Platzman, Jia+]



Have it all: Robust/Incremental Solutions

- Matchings [Hassin-Rubinstein, SIDMA 2002] Given a weighted graph find a single matching and an ordering of its edges such that for every k, the prefix of k edges is near optimal maximum weight matching of size k

- Metric k-median [Mettu-Plaxton, SICOMP 2003] Find an ordering of facilities such that for every k, the prefix of k facilities is a near optimal k-median

Have it all / Best of both: Trade-off two guarantees

Try to find the best possible ratios with respect to the "pessimistic" and "optimistic" extremes

Definition 2 (*Minimization Problems*). Consider two algorithms $\mathcal{P}$ and $\mathcal{O}$ for a minimization problem $\mathcal{I}$. We call an algorithm $\mathcal{H}$, (γ_p, γ_o)*-balanced with respect to algorithms* $\mathcal{P}$ *and* $\mathcal{O}$ if $V_{\mathcal{H}}(I) \leq \min\{\gamma_p \cdot V_{\mathcal{P}}(I), \gamma_o \cdot V_{\mathcal{O}}(I)\}$, for any instance I of the problem.

Note performance is w.r.t. that of the given algorithms but not the "optimal" solutions

Online Algorithms with Uncertain Information
Mahdian, Nazerzadeh & Saberi, EC '07, TALG 2012

Best of both:
Online Resource Allocation

THEOREM 3.1. *Algorithm* $\mathcal{H}(\gamma)$, *for* $\gamma \geq 1$, *is* $\left(\gamma, \frac{\gamma}{(\gamma-1)}\right)$*-balanced with respect to algorithms* $\mathcal{P}$ *and* $\mathcal{O}$. *In addition, suppose algorithm* $\mathcal{O}$ *makes its recommendation based on an optimal allocation for a given estimate. There exists an algorithm* $\mathcal{P}$ *such that for any deterministic* (γ, γ')*-balanced algorithm with respect to* $\mathcal{P}$ *and* $\mathcal{O}$, *we have* $\gamma' \geq \frac{\gamma}{(\gamma-1)}$.

ACM Transactions on Algorithms, Vol. 8, No. 1, Article 2, Publication date: January 2012.

Best of both: AdWords

Maintain (1 – 1/e)-worst case guarantee in the worst case and do quantifiably better for random arrival model

"In this paper we design algorithms that achieve a competitive ratio better than 1 − 1/e on average, while preserving a nearly optimal worst case competitive ratio. Ideally, we want to achieve the best of both worlds, i.e, to design an algorithm with the optimal competitive ratio in both the adversarial and random arrival models. We achieve this for unweighted graphs, but show that it is not possible for weighted graphs."

Simultaneous Approximations for Adversarial and Stochastic Online Budgeted Allocation
Mirrokni, Oveis-Gharan & Zadimoghaddam, SODA '12

Best of both: Balanced guarantees for bandits

The best of both worlds: stochastic and adversarial bandits

Sébastien Bubeck *   Aleksandrs Slivkins†

February 2012

Abstract

We present a bandit algorithm, SAO (Stochastic and Adversarial Optimal), whose regret is, essentially, optimal both for adversarial rewards and for stochastic rewards. Specifically, SAO combines the $O(\sqrt{n})$ worst-case regret of Exp3 [Auer et al., 2002b] for adversarial rewards and the (poly)logarithmic regret of UCB1 [Auer et al., 2002a] for stochastic rewards. Adversarial rewards and stochastic rewards are the two main settings in the literature on (non-Bayesian) multi-armed bandits. Prior work on multi-armed bandits treats them separately, and does not attempt to jointly optimize for both. Our result falls into a general theme of achieving good worst-case performance while also taking advantage of "nice" problem instances, an important issue in the design of algorithms with partially known inputs.

Audience Participation: Other Related Work?

Recall Possible Responses

- ***Relax pessimism:*** Add restrictions to worst-case input scenarios to reduce impact
- ***Temper optimism:*** Make stochastic analysis robust to slight changes in input distribution
- ***"Have it all":*** Build algorithms that achieve the 'best of both worlds' and work for both kinds of inputs
- ***Interpolate Model AND Performance:*** Build hybrid input models interpolating stochastic and adversarial inputs to derive new algorithms that deteriorate smoothly in performance

Proposal: Interpolate Models AND Performance

- ***Model interpolation:*** Input model should allow smooth interpolation between stochastic optimism and worst-case pessimism

- ***Performance interpolation:*** Algorithm should have performance ratio that interpolates smoothly between the better guarantee for stochastic inputs and the worse guarantee for the worst-case

Rest of the talk

[Joint with Guy Blelloch, Kedar Dhamdhere & Suporn Pongnumkul, Summer 2004]

- List Update Problem: Competitive & Average Case Analysis
- A Hybrid Online Model
- Setup and Results from a preliminary experiment
- Conjecture

List Update Problem

Self-organizing sequential search

- Unsorted list

L: $y \rightarrow w \rightarrow z \rightarrow x \rightarrow v \rightarrow u$

- Received a sequence of requests

$$\boldsymbol{\sigma}\textbf{:}\ \sigma(0), \sigma(1), \sigma(2), \sigma(3), \sigma(4), \sigma(5), \sigma(6), \ldots$$

- Cost of accessing the i^{th} element of the list is i. After access, can move it anywhere ahead in the list for free
- Can transpose any pair of adjacent elements at unit cost

List Update Example

L: y → w → z → x → v → u

Action	Cost	Total Cost

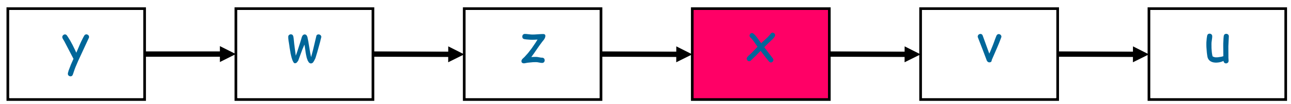
List Update Example

L: y → w → z → x → v → u

Action	Cost	Total Cost
Access x		

List Update Example

L:

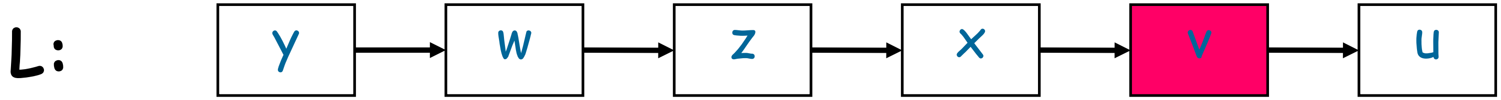


Action	Cost	Total Cost
Access x	4	4

List Update Example

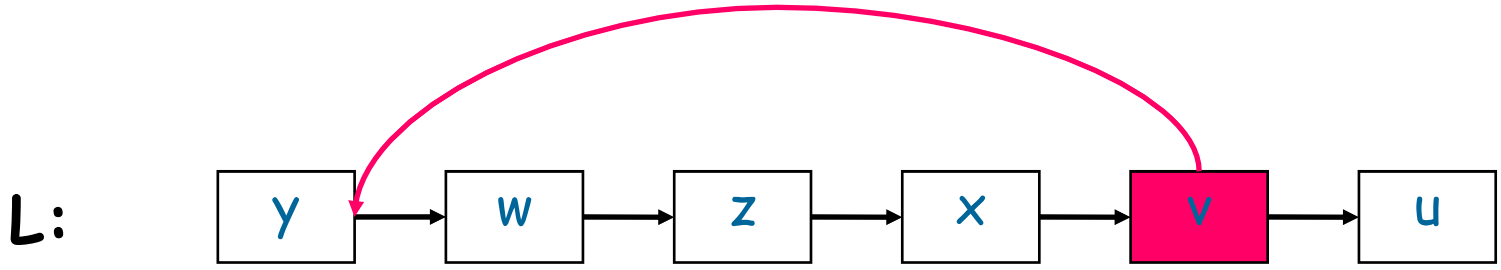
Action	Cost	Total Cost
Access x	4	4
Access v		

List Update Example



Action	Cost	Total Cost
Access x	4	4
Access v	5	9

List Update Example



Action	Cost	Total Cost
Access x	4	4
Access v	5	9
Move v forward to before w		

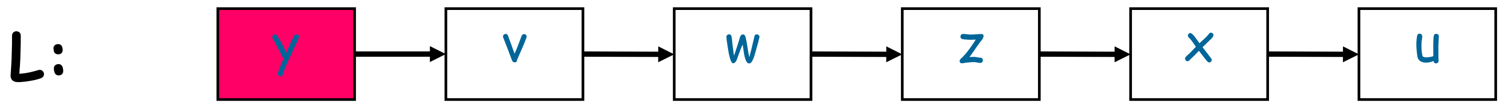
List Update Example

Action	Cost	Total Cost
Access x	4	4
Access v	5	9
Move v forward to before w	0	9

List Update Example

Action	Cost	Total Cost
Access x	4	4
Access v	5	9
Move v forward to before w	0	9
Access y		

List Update Example



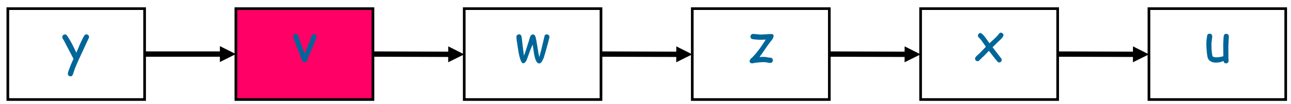
Action	**Cost**	**Total Cost**
Access x	4	4
Access v	5	9
Move v forward to before w	0	9
Access y	1	10

List Update Example

Action	Cost	Total Cost
Access x	4	4
Access v	5	9
Move v forward to before w	0	9
Access y	1	10
Access v		

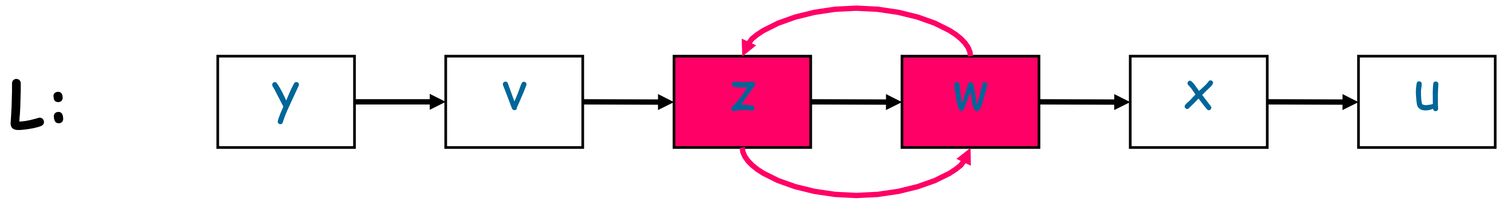
List Update Example

L:



Action	Cost	Total Cost
Access x	4	4
Access v	5	9
Move v forward to before w	0	9
Access y	1	10
Access v	2	12

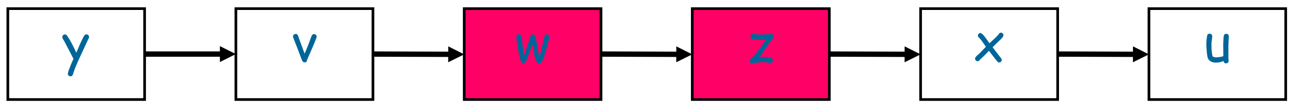
List Update Example



Action	Cost	Total Cost
Access x	4	4
Access v	5	9
Move v forward to before w	0	9
Access y	1	10
Access v	2	12
Transpose w and z		

List Update Example

L:



Action	Cost	Total Cost
Access x	4	4
Access v	5	9
Move v forward to before w	0	9
Access y	1	10
Access v	2	12
Transpose w and z	1	13

Average Case Analysis

- Assume each request comes from a fixed probability distribution, independent of previous requests. Suppose the i^{th} item has probability p_i. Design algorithms to minimize the expected cost.

- Optimal strategy is to keep the list sorted in non-increasing order of p_i.

STAT = Static List

- List is sorted in non-increasing order of the probabilities
- Never moves anything
- Good for when we have a good estimate of the probability distribution.

FC: Frequency Count

- If probability distribution is unknown, estimate it using frequency counts
- Keep list sorted according to counts

Competitive Analysis

- **Definition:** An analysis in which the performance of an online algorithm is compared to the best that could have been achieved if all the inputs had been known in advance.

Competitive Ratio

A:
Our online
algorithm

$C_A(\sigma)$

OPT:
Optimal
Offline
algorithm

$C_{OPT}(\sigma)$

A is c-competitive if $\exists\ a$

$$C_A(\sigma) \le c\ C_{OPT}(\sigma) + a$$

for all request sequences σ

Move-to-Front (MTF)

[Sleator, Tarjan, CACM 1985] When an element is accessed, move it to the front of the list.

- Theorem: MTF has competitive ratio 2 against optimal offline algorithm.

Performance Comparison

- E(FC) /E(STAT) = 1 [Rivest CACM 1976]
- E(MTF)/E(STAT) = $\pi/2$ = 1.58… [Chung, Hajela, Seymour STOC 85]

TS timestamp algorithm

[Albers SODA 95, Albers & Mitzenmacher Algorithmica 1998]

Our work is motivated by the goal to present a universal algorithm that achieves a good competitive ratio but also performs especially well when requests are generated by distributions. ("Have it all")

TS: Insert the requested item, say x, in front of the first item in the list that has been requested at most once since the last request to x. If x has not been requested so far, leave the position of x unchanged.

<u>Theorem</u> TS is 2-competitive

$$E(TS)/E(STAT) = 1.5$$

while for some distribution, $E(MTF)/E(STAT) > 1.57$

New Hybrid Interpolating Model

- Assume a fixed probability distribution

$$\vec{p} = (p_1, p_2, \ldots, p_n)$$

- For each request, with probability ε, let adversary change the request.
- $\varepsilon = 0$ ⇔ Average Case Analysis
- $\varepsilon = 1$ ⇔ Competitive Analysis
- $0 < \varepsilon < 1$ ⇔ Known probability distribution with ε uncertainty.

Desiderata: Interpolating Algorithm for Hybrid Model

- Takes as input estimates of *p* and ε
- Matches best average case performance when ε is low, and matches best competitive ratio when ε is high, and interpolates in between.

Candidate Algorithm: Move-From-Back-Epsilon

- List initially sorted in non-increasing order of probabilities.

- When an element *x* is accessed, promote it past others that have probability up to $p_x + \varepsilon$.

MFBE Example ($\varepsilon = 0.2$)

L: $y \to w \to z \to x \to v \to u$

Prob $p_y = 0.5$ $p_w = 0.2$ $p_z = 0.15$ $p_x = 0.1$ $p_v = 0.03$ $p_u = 0.02$

σ:

Request	Cost	Total Cost

MFBE Example ($\varepsilon = 0.2$)

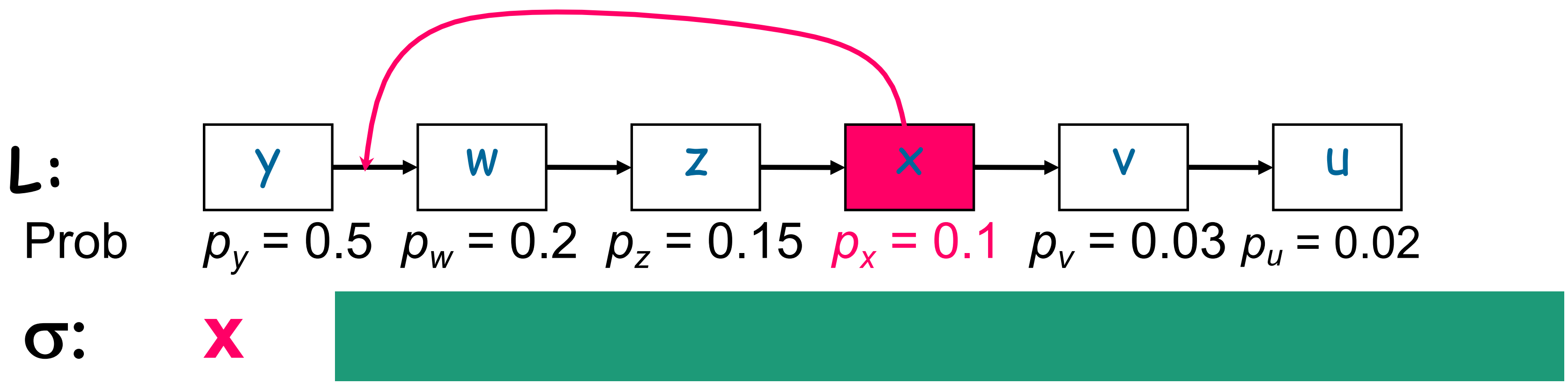
L: y → w → z → x → v → u

Prob $p_y = 0.5$ $p_w = 0.2$ $p_z = 0.15$ $p_x = 0.1$ $p_v = 0.03$ $p_u = 0.02$

σ: **x**

Request	Cost	Total Cost

MFBE Example ($\varepsilon = 0.2$)



Request	Cost	Total Cost
Access x (and move to before w)		

MFBE Example (ε = 0.2)

L: y → x → w → z → v → u

Prob $p_y = 0.5$ $p_x = 0.1$ $p_w = 0.2$ $p_z = 0.15$ $p_v = 0.03$ $p_u = 0.02$

σ: **x**

Request	Cost	Total Cost
Access x (and move to before w)	4	4

MFBE Example ($\varepsilon = 0.2$)

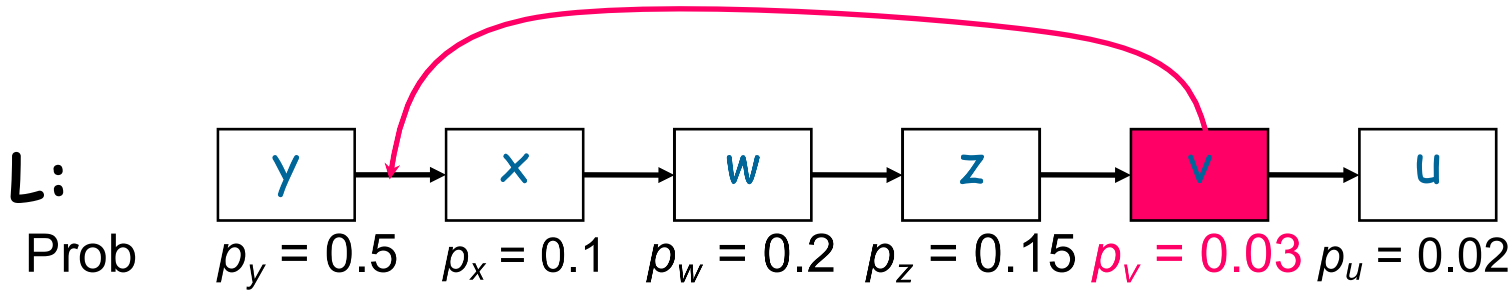
L: y → x → w → z → v → u

Prob $p_y = 0.5$ $p_x = 0.1$ $p_w = 0.2$ $p_z = 0.15$ $p_v = 0.03$ $p_u = 0.02$

σ: **x** **v**

Request	Cost	Total Cost
Access x (and move to before w)	4	4

MFBE Example (ε = 0.2)



σ: **x** **v**

Request	Cost	Total Cost
Access x (and move to before w)	4	4
Access v (and move to before x)		

MFBE Example ($\varepsilon = 0.2$)

L: y → v → x → w → z → u

Prob $p_y = 0.5$ $p_v = 0.03$ $p_x = 0.1$ $p_w = 0.2$ $p_z = 0.15$ $p_u = 0.02$

σ: **x** **v**

Request	Cost	Total Cost
Access x (and move to before w)	4	4
Access v (and move to before x)	5	9

MFBE Example ($\varepsilon = 0.2$)

L: y → v → x → w → z → u

Prob $p_y = 0.5$ $p_v = 0.03$ $p_x = 0.1$ $p_w = 0.2$ $p_z = 0.15$ $p_u = 0.02$

σ: **x** **v** **y**

Request	Cost	Total Cost
Access x (and move to before w)	4	4
Access v (and move to before x)	5	9

MFBE Example ($\varepsilon = 0.2$)

L: y → v → x → w → z → u

Prob $p_y = 0.5$ $p_v = 0.03$ $p_x = 0.1$ $p_w = 0.2$ $p_z = 0.15$ $p_u = 0.02$

σ: **x** **v** **y**

Request	Cost	Total Cost
Access x (and move to before w)	4	4
Access v (and move to before x)	5	9
Access y		

MFBE Example ($\varepsilon = 0.2$)

L: y → v → x → w → z → u

Prob $p_y = 0.5$ $p_v = 0.03$ $p_x = 0.1$ $p_w = 0.2$ $p_z = 0.15$ $p_u = 0.02$

σ: **x** **v** **y**

Request	Cost	Total Cost
Access x (and move to before w)	4	4
Access v (and move to before x)	5	9
Access y	1	10

MFBE Example ($\varepsilon = 0.2$)

L: y → v → x → w → z → u

Prob $p_y = 0.5$ $p_v = 0.03$ $p_x = 0.1$ $p_w = 0.2$ $p_z = 0.15$ $p_u = 0.02$

σ: **x** **v** **y** **v**

Request	Cost	Total Cost
Access x (and move to before w)	4	4
Access v (and move to before x)	5	9
Access y	1	10

MFBE Example ($\varepsilon = 0.2$)

L: y → v → x → w → z → u

Prob $p_y = 0.5$ $p_v = 0.03$ $p_x = 0.1$ $p_w = 0.2$ $p_z = 0.15$ $p_u = 0.02$

σ: x v y v

Request	Cost	Total Cost
Access x (and move to before w)	4	4
Access v (and move to before x)	5	9
Access y	1	10
Access v		

MFBE Example ($\varepsilon = 0.2$)

L: y → v → x → w → z → u

Prob $p_y = 0.5$ $p_v = 0.03$ $p_x = 0.1$ $p_w = 0.2$ $p_z = 0.15$ $p_u = 0.02$

σ: x v y v

Request	Cost	Total Cost
Access x (and move to before w)	4	4
Access v (and move to before x)	5	9
Access y	1	10
Access v	2	12

Difficulties in Proving Properties of MFBE

- Must compare with OPT rather than STAT
- OPT can be computed by Dynamic programming
  - Trivial way = $\mathrm{O}((n!)^2 m)$
  - Improvement = $\mathrm{O}((2^n)(n!)m)$

    [Reingold, Westbrook, 1996]
- Hard to derive useful properties of OPT

Experiments (Pongnumkul ALADDIN REU 2004)

- Motivation: To see the behavior of algorithms in our hybrid model.

- Measurement: We measure the performance of an online algorithm by the average competitive ratio.

Our Experiment

- Variables in our experiment
  - Type of List Update Algorithm (MTF, STAT, MFBE)
  - Type of Probability Distribution
  - Type of Adversary
  - Epsilon: ε

Our Experiment

- We generate a request sequence of length 100, with a chosen probability distribution
- Then, with probability ε change the request sequence adverserially

Our Experiment

- Record the cost incurred by the online algorithm = $Cost_A(\sigma)$
- Use Dynamic Programming to find optimum cost of that request sequence = $Cost_{OPT}(\sigma)$.
- Competitive Ratio = $Cost_A(\sigma)/Cost_{OPT}(\sigma)$
- Repeat this 100 times to find the average competitive ratio.

Distribution

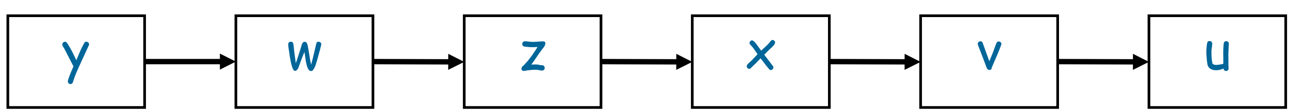
- Geometric Distribution:
  - P[i] / $1/2^i$
- Uniform Distribution:
  - P[i] = 1/n, n = length of the list
- Zipfian Distribution (Zipf(2)):
  - P[i] / $1/i^2$

Cruel Adversary

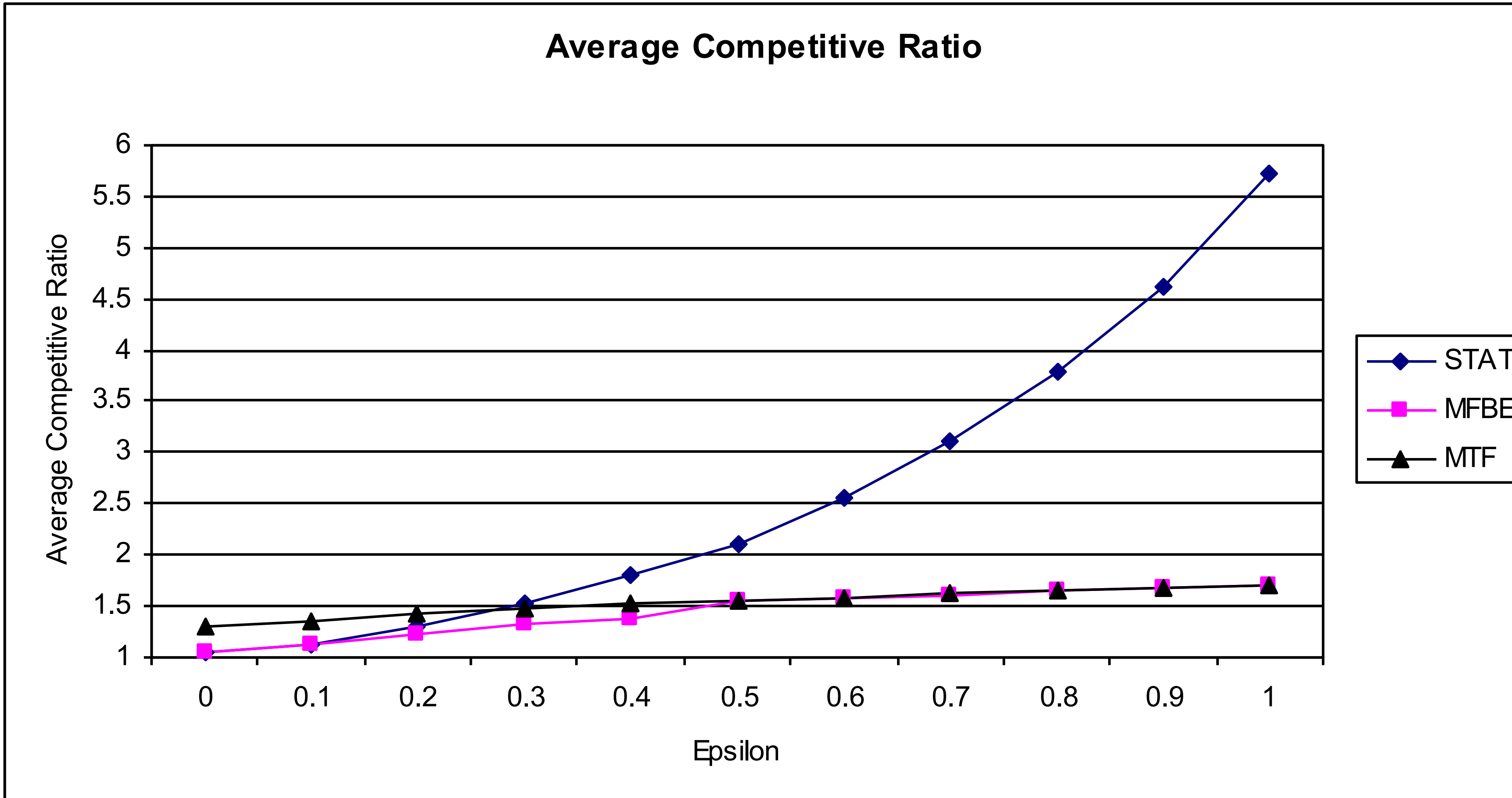
- This is an adaptive adversary

- Looks at the current list and request the last item in the list.

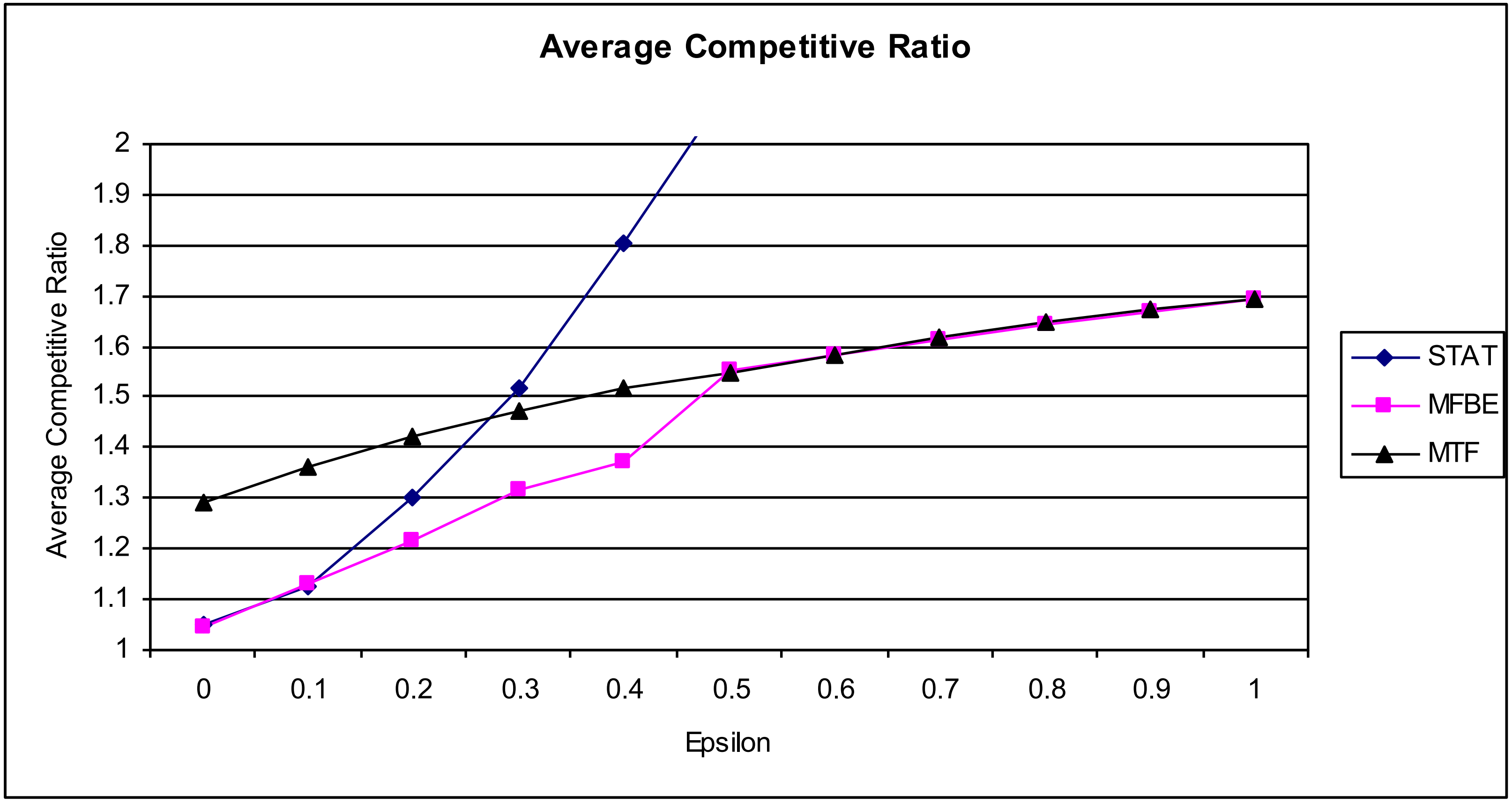
L:



Cruel Adversary, Geometric Distribution, n=6



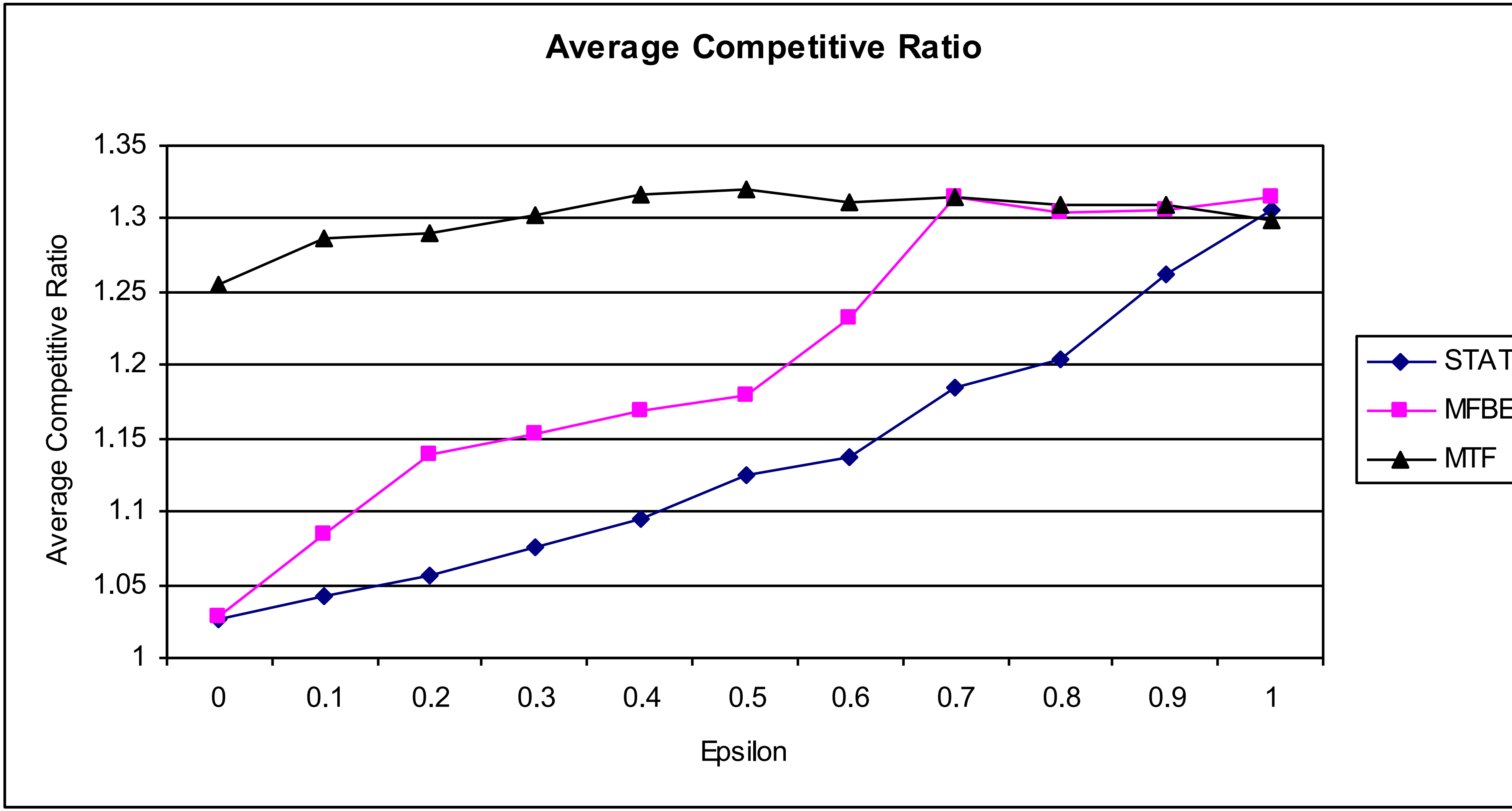
Cruel Adversary, Geometric Distribution, n=6 (zoomed in)



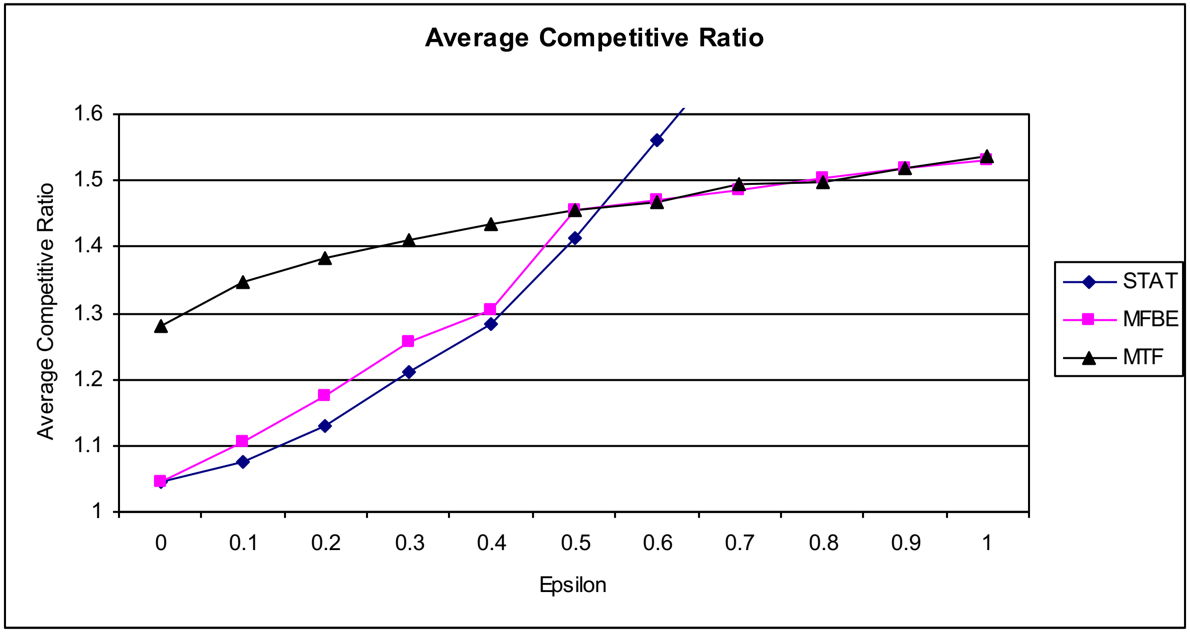
Other Adversaries

- ***Geometric adversary*** chooses elements randomly, according to the geometric distribution on the reversed STAT order
- ***Uniform adversary*** requests elements from the list uniformly at random
- ***Oblivious Adversary*** doesn’t look at the current list

Uniform Adversary, Zipfian2 Distribution, n=6



Reversed Geometric Adversary, Geometric Distribution, n=6



Observations

- The performance of any algorithm in this hybrid model depends heavily on the type of adversary.
- MFBE seems better than the worse of STAT and MTF.

Conjecture

(Not the worst) The average competitive ratio of MFBE is dominated by the maximum of the average competitive ratios of STAT and MTF.

- ***(Best of all)*** What we wanted but probably not true: Avg competitive ratio of MFBE is at most the minimum of the average competitive ratios of STAT and MTF.

Summary

Interpolate Model AND Performance: Build hybrid input models interpolating stochastic and adversarial inputs to derive new algorithms that deteriorate smoothly in performance

- Prove/disprove “MFBE is not the worst”
- Find a setting where the hybrid model interpolates stochastic and worst-case inputs, and the algorithm interpolates the performance of average case and worst case algorithms
- Are there more effective approaches to trade-off stochastic optimism and worst-case pessimism?