Efficient Algorithms for Online Decision Problems

Dave Buchfuhrer

January 15, 2009

The Model

- In this model, we have n experts

The Model

- In this model, we have n experts

e_1	e_2	e_3	e_4

The Model

- In this model, we have n experts
- Every round, we must pick an expert

e_1	e_2	e_3	e_4

The Model

- In this model, we have n experts
- Every round, we must pick an expert

e_1	e_2	e_3	e_4
✓			

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed

e_1	e_2	e_3	e_4
✓			

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed

e_1	e_2	e_3	e_4
.2	.5	.1	.8

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed

e_1	e_2	e_3	e_4
.2	.5	.1	.8
		✓	

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed

e_1	e_2	e_3	e_4
.2	.5	.1	.8
.5	.3	.6	0

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed

e_1	e_2	e_3	e_4
.2	.5	.1	.8
.5	.3	.6	0
			✓

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed

e_1	e_2	e_3	e_4
.2	.5	.1	.8
.5	.3	.6	0
.9	.4	.2	.3

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed

e_1	e_2	e_3	e_4
.2	.5	.1	.8
.5	.3	.6	0
.9	.4	.2	.3
		✓	

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed

e_1	e_2	e_3	e_4
.2	.5	.1	.8
.5	.3	.6	0
.9	.4	.2	.3
.1	.6	.8	.9

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed
- The goal is to minimize the total cost incurred

e_1	e_2	e_3	e_4
.2	.5	.1	.8
.5	.3	.6	0
.9	.4	.2	.3
.1	.6	.8	.9

The Model

- In this model, we have n experts
- Every round, we must pick an expert
- After this choice, the cost of each expert is revealed
- The goal is to minimize the total cost incurred

e_1	e_2	e_3	e_4
.2	.5	.1	.8
.5	.3	.6	0
.9	.4	.2	.3
.1	.6	.8	.9

Total cost: 1.9

Limit to Single Expert

e_1	e_2	e_3	e_4

Limit to Single Expert

e_1	e_2	e_3	e_4
1	0	1	1

Limit to Single Expert

e_1	e_2	e_3	e_4
1	0	1	1
1	1	1	0

Limit to Single Expert

e_1	e_2	e_3	e_4
1	0	1	1
1	1	1	0
0	1	1	1

Limit to Single Expert

e_1	e_2	e_3	e_4
1	0	1	1
1	1	1	0
0	1	1	1
1	0	1	1

Purely Random Strategies are Bad

e_1	e_2	e_3	e_4
1	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0

Purely Random Strategies are Bad

e_1	e_2	e_3	e_4
1	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0

Purely Random Strategies are Bad

e_1	e_2	e_3	e_4
1	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0

Purely Random Strategies are Bad

e_1	e_2	e_3	e_4
1	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0

Purely Random Strategies are Bad

e_1	e_2	e_3	e_4
1	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0

Following the Best Track Record

e_1	e_2	e_3	e_4

Following the Best Track Record

e_1	e_2	e_3	e_4
☺			

Following the Best Track Record

e_1	e_2	e_3	e_4	
1	0	0	0	☹

Following the Best Track Record

e_1	e_2	e_3	e_4	
1	0	0	0	☹
	☺			

Following the Best Track Record

e_1	e_2	e_3	e_4	
1	0	0	0	☹
0	1	0	0	☹

Following the Best Track Record

e_1	e_2	e_3	e_4	
1	0	0	0	☹
0	1	0	0	☹
		☺		

Following the Best Track Record

e_1	e_2	e_3	e_4	
1	0	0	0	☹
0	1	0	0	☹
0	0	1	0	☹

I'm feeling good about this one!

e_1	e_2	e_3	e_4	
1	0	0	0	☹
0	1	0	0	☹
0	0	1	0	☹
			☺	

Damnit!

e_1	e_2	e_3	e_4	
1	0	0	0	☹
0	1	0	0	☹
0	0	1	0	☹
0	0	0	1	☹

Failure of Follow the Leader

At each step t in follow the leader, we can

1. Pick the expert with the best total so far
2. Fail to do so

Failure of Follow the Leader

At each step t in follow the leader, we can

1. Pick the expert with the best total so far
2. Fail to do so

Case 1: we increase our total cost by at most the same amount as the best strategy

Failure of Follow the Leader

At each step t in follow the leader, we can

1. Pick the expert with the best total so far
2. Fail to do so

Case 1: we increase our total cost by at most the same amount as the best strategy
Case 2: we increase our total cost by at most 1 more than the cost increase of the best strategy

Example

e_1	e_2	e_3	e_4	guess	leader

Example

e_1	e_2	e_3	e_4	guess	leader
✓					

Example

e_1	e_2	e_3	e_4	guess	leader
.2	.5	1	.5	e_1 (.2)	e_1 (.2)

Example

e_1	e_2	e_3	e_4	guess	leader
.2	.5	1	.5	e_1 (.2)	e_1 (.2)
✓					

Example

e_1	e_2	e_3	e_4	guess	leader
.2	.5	1	.5	e_1 (.2)	e_1 (.2)
.7	.2	.3	.1	e_1 (.9)	e_4 (.6)

Example

e_1	e_2	e_3	e_4	guess	leader
.2	.5	1	.5	e_1 (.2)	e_1 (.2)
.7	.2	.3	.1	e_1 (.9)	e_4 (.6)
			✓		

Example

e_1	e_2	e_3	e_4	guess	leader
.2	.5	1	.5	e_1 (.2)	e_1 (.2)
.7	.2	.3	.1	e_1 (.9)	e_4 (.6)
.3	.6	.8	1	e_4 (1.9)	e_1 (1.2)

Example

e_1	e_2	e_3	e_4	guess	leader
.2	.5	1	.5	e_1 (.2)	e_1 (.2)
.7	.2	.3	.1	e_1 (.9)	e_4 (.6)
.3	.6	.8	1	e_4 (1.9)	e_1 (1.2)
✓					

Example

e_1	e_2	e_3	e_4	guess	leader
.2	.5	1	.5	e_1 (.2)	e_1 (.2)
.7	.2	.3	.1	e_1 (.9)	e_4 (.6)
.3	.6	.8	1	e_4 (1.9)	e_1 (1.2)
.1	.6	.4	0	e_1 (2.0)	e_1 (1.3)

Example

e_1	e_2	e_3	e_4	guess	leader
.2	.5	1	.5	e_1 (.2)	e_1 (.2)
.7	.2	.3	.1	e_1 (.9)	e_4 (.6)
.3	.6	.8	1	e_4 (1.9)	e_1 (1.2)
.1	.6	.4	0	e_1 (2.0)	e_1 (1.3)
✓					

Example

e_1	e_2	e_3	e_4	guess	leader
.2	.5	1	.5	e_1 (.2)	e_1 (.2)
.7	.2	.3	.1	e_1 (.9)	e_4 (.6)
.3	.6	.8	1	e_4 (1.9)	e_1 (1.2)
.1	.6	.4	0	e_1 (2.0)	e_1 (1.3)
.5	.2	.3	.4	e_1 (2.5)	e_1 (1.8)

Reason for Failure

So the total cost of Follow the Leader is at most

$$\text{best cost} + \#\ \text{times leader guess was wrong}$$

Reason for Failure

So the total cost of Follow the Leader is at most

$$\text{best cost} + \#\text{ times leader guess was wrong}$$

or in other words,

$$\text{final leader's cost} + \#\text{ times the leader guess changed}$$

k-Armed Bandit Connection

- Confidence intervals helped with k-armed bandits

k-Armed Bandit Connection

- Confidence intervals helped with k-armed bandits
- Here, we'll just fudge the numbers to prevent leader changes

k-Armed Bandit Connection

- Confidence intervals helped with k-armed bandits
- Here, we'll just fudge the numbers to prevent leader changes
- We add a random perturbation $pert[i]$ to each expert i

Adding Randomness

e_1	e_2	e_3	e_4
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Adding Randomness

e_1	e_2	e_3	e_4
3	10	2	8
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Too Much Randomness?

e_1	e_2	e_3	e_4
3	10	2	8
1	0	1	0
1	0	1	0
1	0	1	0
1	0	1	0

Too Much Randomness?

e_1	e_2	e_3	e_4
3	10	2	8
1	0	1	0
1	0	1	0
1	0	1	0
1	0	1	0

Getting it Right

In order to do well, we add a random variable to each expert with exponential density function

$$\epsilon e^{\epsilon x}$$

for negative perturbations x

Getting it Right

In order to do well, we add a random variable to each expert with exponential density function

$$\epsilon e^{\epsilon x}$$

for negative perturbations x

We hope that

- The expected number of leader changes is small compared to the final leader cost

Getting it Right

In order to do well, we add a random variable to each expert with exponential density function

$$\epsilon e^{\epsilon x}$$

for negative perturbations x

We hope that

- The expected number of leader changes is small compared to the final leader cost
- The final leader cost is close to the min cost

Number of Leader Changes

We wish to show that

$$E[\#\ \text{changes of leader}] \le \epsilon E[\text{total cost}]$$

Number of Leader Changes

We wish to show that

$$E[\#\text{ changes of leader}] \leq \epsilon E[\text{total cost}]$$

which shows us that

$$E[\text{total cost}] \leq E[\text{final leader cost}] + \epsilon E[\text{total cost}]$$

Number of Leader Changes

We wish to show that

$$E[\#\text{ changes of leader}] \leq \epsilon E[\text{total cost}]$$

which shows us that

$$E[\text{total cost}] \leq E[\text{final leader cost}] + \epsilon E[\text{total cost}]$$

giving us

$$E[\text{total cost}] \leq \frac{1}{1-\epsilon} E[\text{final leader cost}]$$

Chance of Changing Leader

- If expert i is the current leader, consider his current costs, as compared to the costs of all other experts, as well as their perturbations

Chance of Changing Leader

- If expert i is the current leader, consider his current costs, as compared to the costs of all other experts, as well as their perturbations
- Given this info, i must have a sufficiently small perturbation to be leader

Chance of Changing Leader

- If expert i is the current leader, consider his current costs, as compared to the costs of all other experts, as well as their perturbations
- Given this info, i must have a sufficiently small perturbation to be leader
- Since the exponential distribution is memoryless, the chances that it's c smaller than necessary only depend on c

Chance of Changing Leader

- If expert i is the current leader, consider his current costs, as compared to the costs of all other experts, as well as their perturbations
- Given this info, i must have a sufficiently small perturbation to be leader
- Since the exponential distribution is memoryless, the chances that it's c smaller than necessary only depend on c
- This chance happens to be greater than $1 - \epsilon c$

Leader Change

- So there's only an ϵc chance of the leader being leader by less than a margin of c

Leader Change

- So there's only an ϵc chance of the leader being leader by less than a margin of c
- Let c_t be the current leader's next cost at time t
- $\sum_t c_t = \text{total cost}$

Leader Change

- So there's only an ϵc chance of the leader being leader by less than a margin of c
- Let c_t be the current leader's next cost at time t
- $\sum_t c_t = \text{total cost}$
- So total number of changes is $\epsilon(\text{total cost})$

Final Leader Cost

This leaves us with the need to bound $E[\text{final leader cost}]$, as the final leader is not necessarily optimal

Final Leader Cost

This leaves us with the need to bound $E[\text{final leader cost}]$, as the final leader is not necessarily optimal

- Our leader can only be as much worse as the biggest perturbation

Final Leader Cost

This leaves us with the need to bound $E[\text{final leader cost}]$, as the final leader is not necessarily optimal

- Our leader can only be as much worse as the biggest perturbation
- Because the distribution is exponential, the expected max perturbation grows logarithmically

Final Leader Cost

This leaves us with the need to bound $E[\text{final leader cost}]$, as the final leader is not necessarily optimal

- Our leader can only be as much worse as the biggest perturbation
- Because the distribution is exponential, the expected max perturbation grows logarithmically
- In particular, we get a bound of $(1 + \ln n)/\epsilon$

Tying it Together

Combining the bounds on the number of wrong guesses with the bound on the error in our final guess, we get

$$E[\text{total cost}](1-\epsilon) \leq \text{min cost} + \frac{\ln n}{\epsilon}$$

Tying it Together

Combining the bounds on the number of wrong guesses with the bound on the error in our final guess, we get

$$E[\text{total cost}](1-\epsilon) \leq \text{min cost} + \frac{\ln n}{\epsilon}$$

which shows an interesting tradeoff between ϵ and $1-\epsilon$ when balancing the amount of randomness

Refreshing the Randomness

e_1	e_2	e_3	e_4
8	8	6	7

Refreshing the Randomness

e_1	e_2	e_3	e_4
9	3	6	4
1	0	1	0

Refreshing the Randomness

e_1	e_2	e_3	e_4
0	3	2	1
1	0	1	0
1	0	1	0

Refreshing the Randomness

e_1	e_2	e_3	e_4
6	2	1	4
1	0	1	0
1	0	1	0
1	0	1	0

Refreshing the Randomness

e_1	e_2	e_3	e_4
1	0	1	0
1	0	1	0
1	0	1	0
1	0	1	0

Linear Generalization

- Fix some $D \subset \mathbb{R}^n$

Linear Generalization

- Fix some $D \subset \mathbb{R}^n$
- At time t, choose some $d_t \in D$

Linear Generalization

- Fix some $D \subset \mathbb{R}^n$
- At time t, choose some $d_t \in D$
- After d_t is chosen, a vector s_t is revealed

Linear Generalization

- Fix some $D \subset \mathbb{R}^n$
- At time t, choose some $d_t \in D$
- After d_t is chosen, a vector s_t is revealed
- The cost incurred is $d_t \cdot s_t$

Linear Generalization

- Fix some $D \subset \mathbb{R}^n$
- At time t, choose some $d_t \in D$
- After d_t is chosen, a vector s_t is revealed
- The cost incurred is $d_t \cdot s_t$
- We wish to compete with the best fixed choice $d_t = d \; \forall t$

Linear Generalization

- Fix some $D \subset \mathbb{R}^n$
- At time t, choose some $d_t \in D$
- After d_t is chosen, a vector s_t is revealed
- The cost incurred is $d_t \cdot s_t$
- We wish to compete with the best fixed choice $d_t = d \; \forall t$
- In the 4-player expert case,

  $$D = (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)$$

  and the s_t are the cost vectors

Algorithm for Linear Generalization

With this generalization, the same algorithm works:

- Choose a random vector p_t
- Find the $d \in D$ that minimizes $d \cdot p_t + \sum_i d \cdot s_i$ and choose it

Other Problems in this Framework

The linear generalization covers many interesting online optimization problems, including online shortest path:

Other Problems in this Framework

The linear generalization covers many interesting online optimization problems, including online shortest path:

- We are given a graph with 2 labeled vertices s and t

Other Problems in this Framework

The linear generalization covers many interesting online optimization problems, including online shortest path:

- We are given a graph with 2 labeled vertices s and t
- Every round, we pick a path from s to t

Other Problems in this Framework

The linear generalization covers many interesting online optimization problems, including online shortest path:

- We are given a graph with 2 labeled vertices s and t
- Every round, we pick a path from s to t
- Afterward, all edge weights are revealed

Other Problems in this Framework

The linear generalization covers many interesting online optimization problems, including online shortest path:

- We are given a graph with 2 labeled vertices s and t
- Every round, we pick a path from s to t
- Afterward, all edge weights are revealed
- We wish to minimize the sum of all path lengths

Other Problems in this Framework

The linear generalization covers many interesting online optimization problems, including online shortest path:

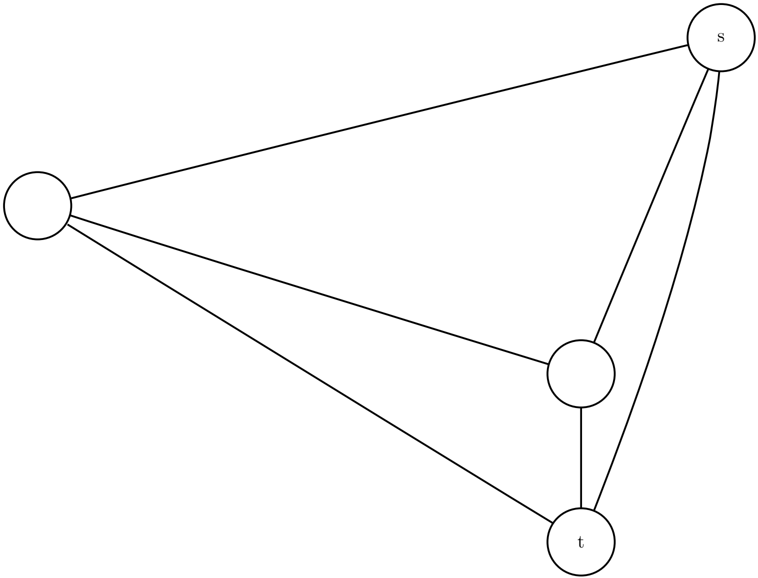
- We are given a graph with 2 labeled vertices s and t
- Every round, we pick a path from s to t
- Afterward, all edge weights are revealed
- We wish to minimize the sum of all path lengths
- We are competing against the optimal fixed path choice

Other Problems in this Framework

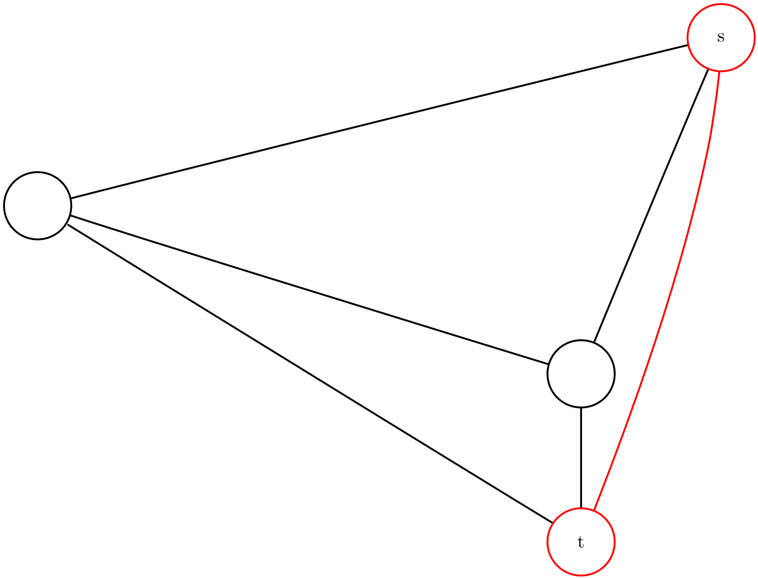
The linear generalization covers many interesting online optimization problems, including online shortest path:

- We are given a graph with 2 labeled vertices s and t
- Every round, we pick a path from s to t
- Afterward, all edge weights are revealed
- We wish to minimize the sum of all path lengths
- We are competing against the optimal fixed path choice
- Here $d \in D$ is a vector indicating the edges contained in a path, and s_t represents the edge weights

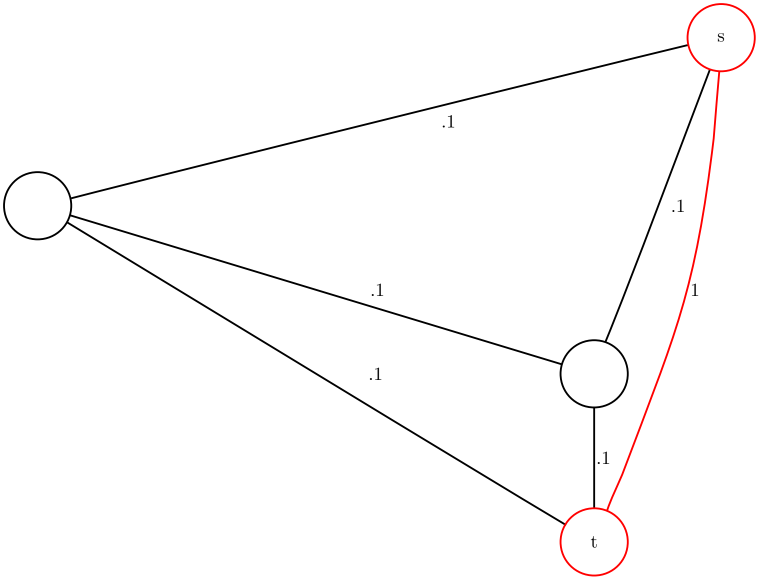
Online Shortest Paths Example



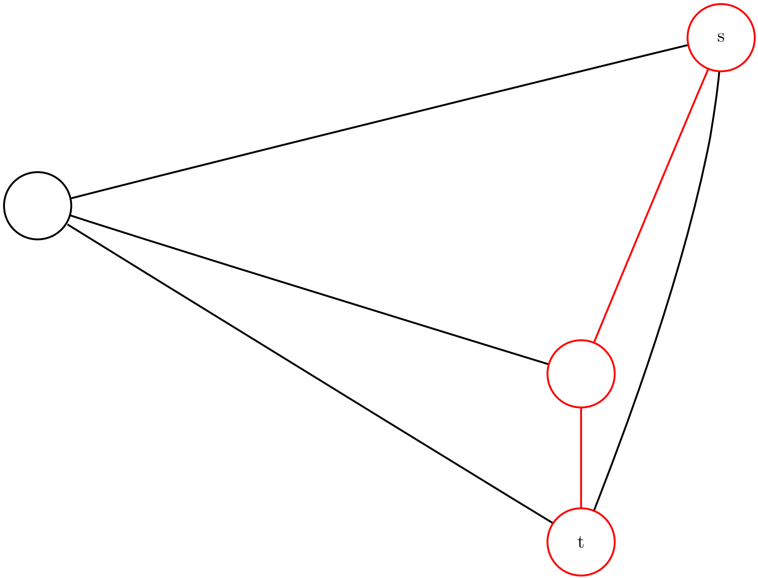
Online Shortest Paths Example



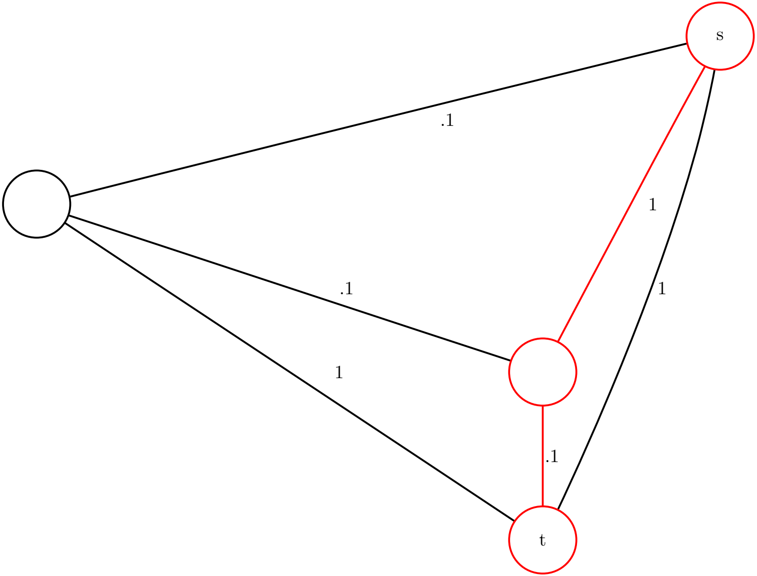
Online Shortest Paths Example



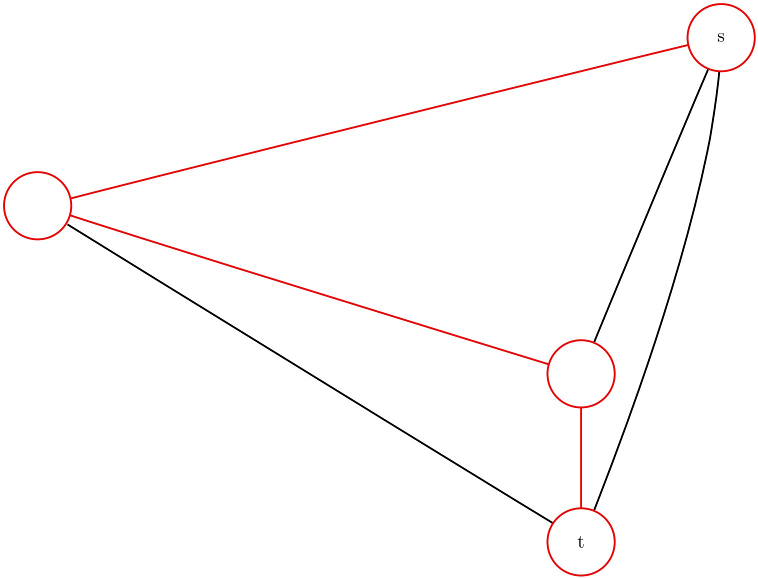
Online Shortest Paths Example



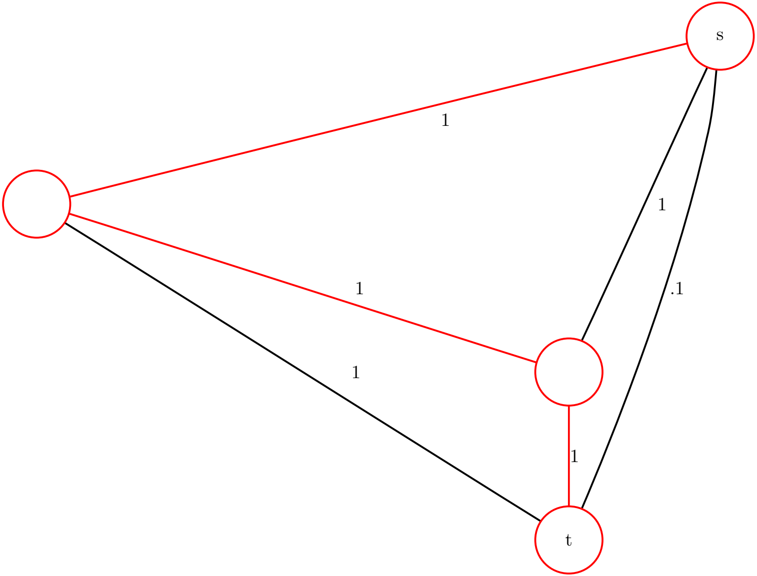
Online Shortest Paths Example



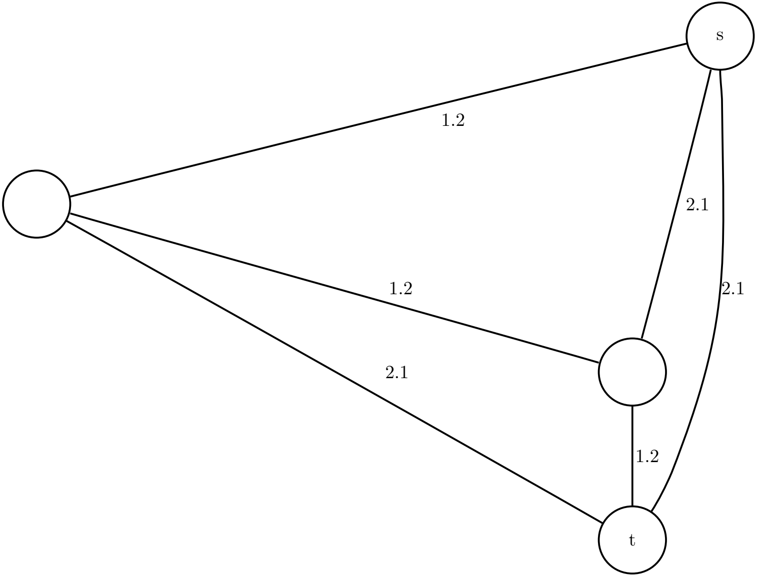
Online Shortest Paths Example



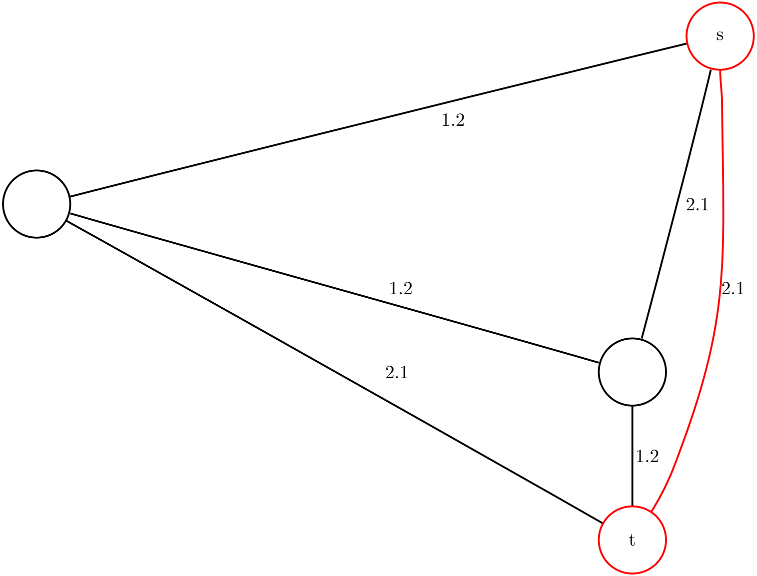
Online Shortest Paths Example



Follow the Leader



Follow the Leader



Any Questions?