Search and Rediscovery

Martino Banchio

Google Research

Joint work with **Suraj Malladi**

June 22nd, 2024

Research vs homework

Both unfamiliar environments for searchers

Both involve trial-and-error to find more promising approaches

But question is *open* at the time of research

What does the process of re-discovery look like?

Trade secrets

Leader develops innovation

Keeps method as trade secret

Competitors learns such an innovation is feasible

How does this affect their R&D?

Behavioral theory of the firm

Problemistic search (Cyert and March 1963; Simon 1962)

“Search within the firm is problem-oriented…innovation by a competitor”

Rugged landscapes (Levinthal 1997; Bellinger et al. 2012)

Mapping from choices to performance is complex and unpredictable

This paper

Model of “problemistic search on rugged landscapes”

Characterize the optimal search process

Outline

1. Model
2. Simple search policies
3. Proof ideas
4. Discussion

States

$S \equiv [0, 1]$ is the **search space**

$Q \subset \mathbb{R}_+^S$ is the state space

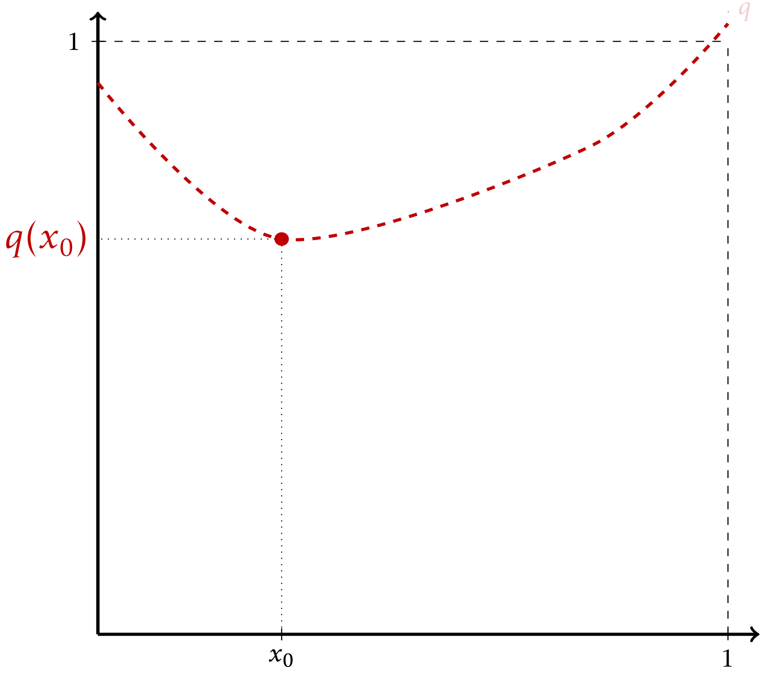
An element $q \in Q$ is a **quality index**

Actions

x_t: searcher's time t action

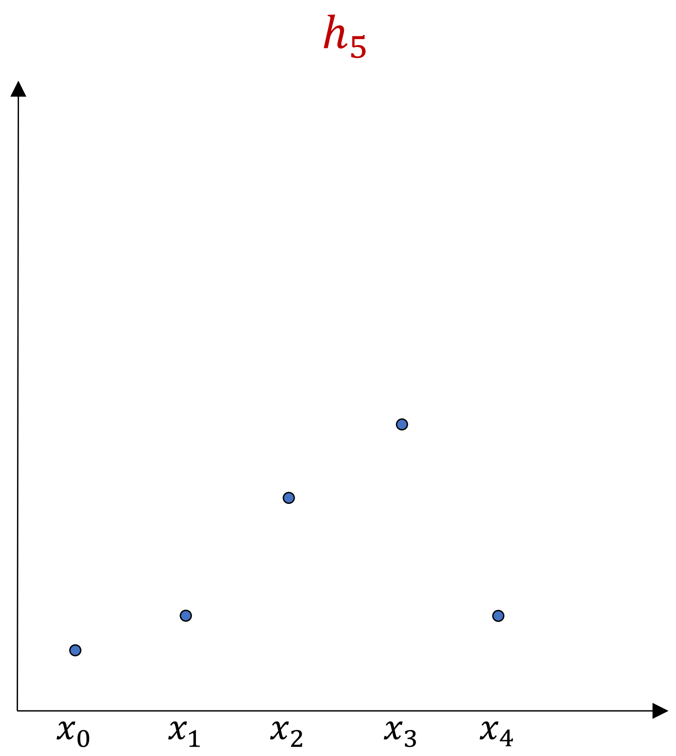
$x_t \in S$: learns $q(x_t)$ in state $q \in Q$

$x_t = \emptyset$: search ends, payoff realized



Learning

$h_t = ((x_i, z_i))_{i=0}^{t-1}$ is a time t history, with $z_i = q(x_i)$

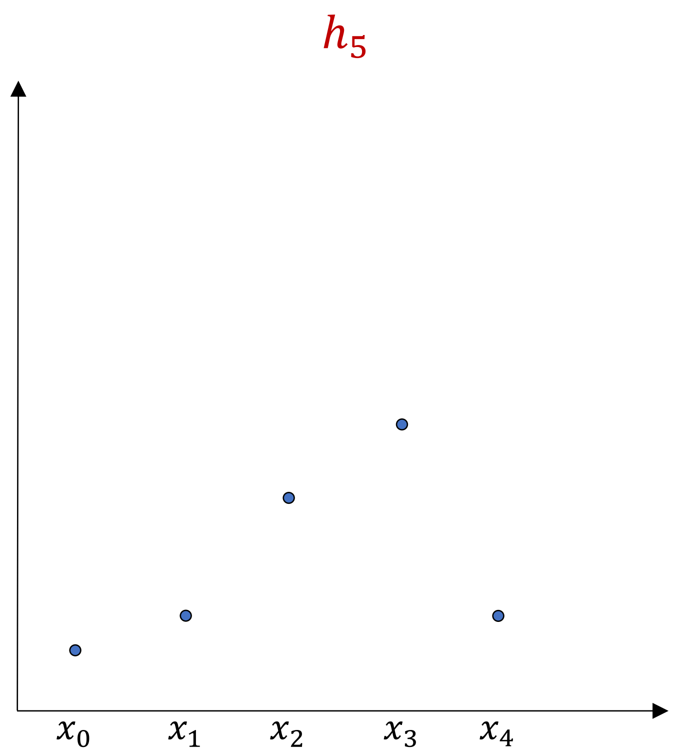


Learning

$h_t = ((x_i, z_i))_{i=0}^{t-1}$ is a time t history, with $z_i = q(x_i)$

$Q_h \subset Q$ is set of quality indices **consistent** with h

$$Q_h \equiv \{q \in Q | q(x) = z, \forall (x, z) \in h\}$$



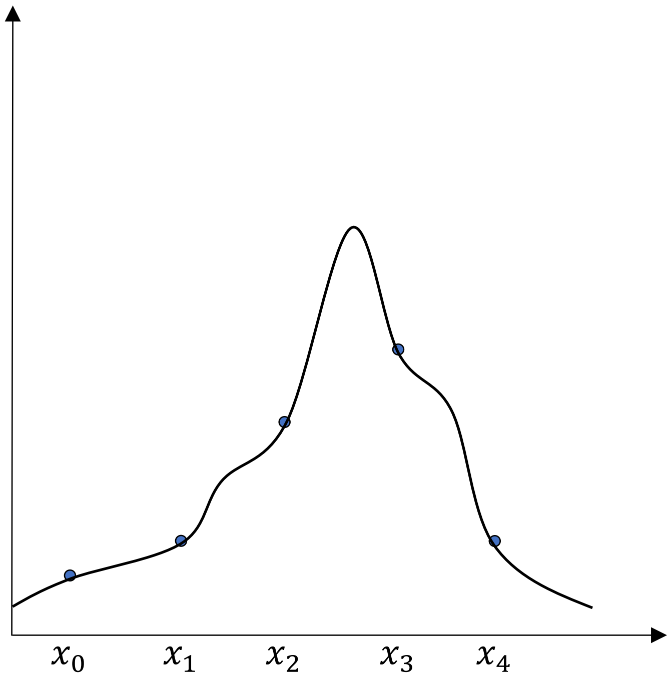
Learning

$h_t = ((x_i, z_i))_{i=0}^{t-1}$ is a time t history, with $z_i = q(x_i)$

$Q_h \subset Q$ is set of quality indices **consistent** with h

$$Q_h \equiv \{q \in Q | q(x) = z, \forall (x, z) \in h\}$$

consistent q



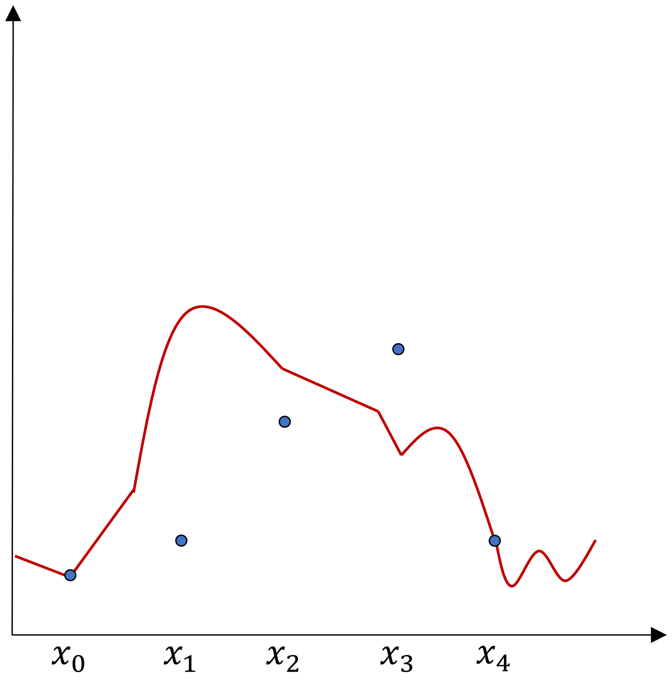
Learning

$h_t = ((x_i, z_i))_{i=0}^{t-1}$ is a time t history, with $z_i = q(x_i)$

$Q_h \subset Q$ is set of quality indices **consistent** with h

$$Q_h \equiv \{q \in Q | q(x) = z, \forall (x, z) \in h\}$$

inconsistent q



Learning

$h_t = ((x_i, z_i))_{i=0}^{t-1}$ is a time t history, with $z_i = q(x_i)$

$Q_h \subset Q$ is set of quality indices **consistent** with h

$$Q_h \equiv \{q \in Q | q(x) = z, \forall (x, z) \in h\}$$

H is set of all histories h where Q_h nonempty

Rugged landscapes and rediscovery

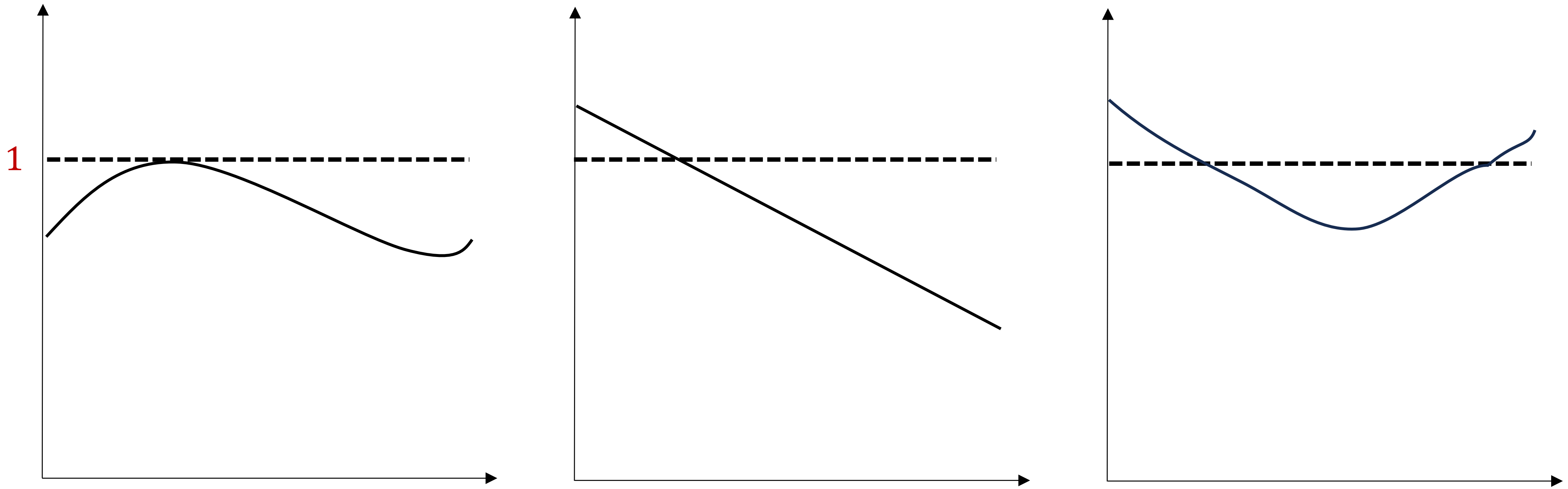
Let $L \in \mathbb{R}_+$ (ruggedness) and $k \in \mathbb{R}_+$ (target)

Assumption:

$Q = \{q \in \mathbb{R}_+^S \mid q$ is L–Lipschitz continuous **and** $\exists x \in S$ s.t. $q(x) = k\}$

This talk: $L = 1$ and $k = 1$

Rugged landscapes and rediscovery



Payoffs

Payoff to ending search at empty history h_0 is 0

Payoff to ending search at any other $h_t \in H$:

$$p(h_t) = \max_{i \in \{0,\dots,t-1\}} z_i - c \cdot t$$

Search strategies

A **strategy** is a map $\sigma: H \rightarrow S \cup \{\emptyset\}$

$h \in H$ is a **terminal history** for strategy σ if $\sigma(h) = \emptyset$

σ **terminates** if it reaches a terminal history h_q^σ from any $h \in H, q \in Q_h$

Σ is set of all strategies that terminate

Solution

σ^* is (ex-ante) **optimal** if at the empty history $h = h_0$:

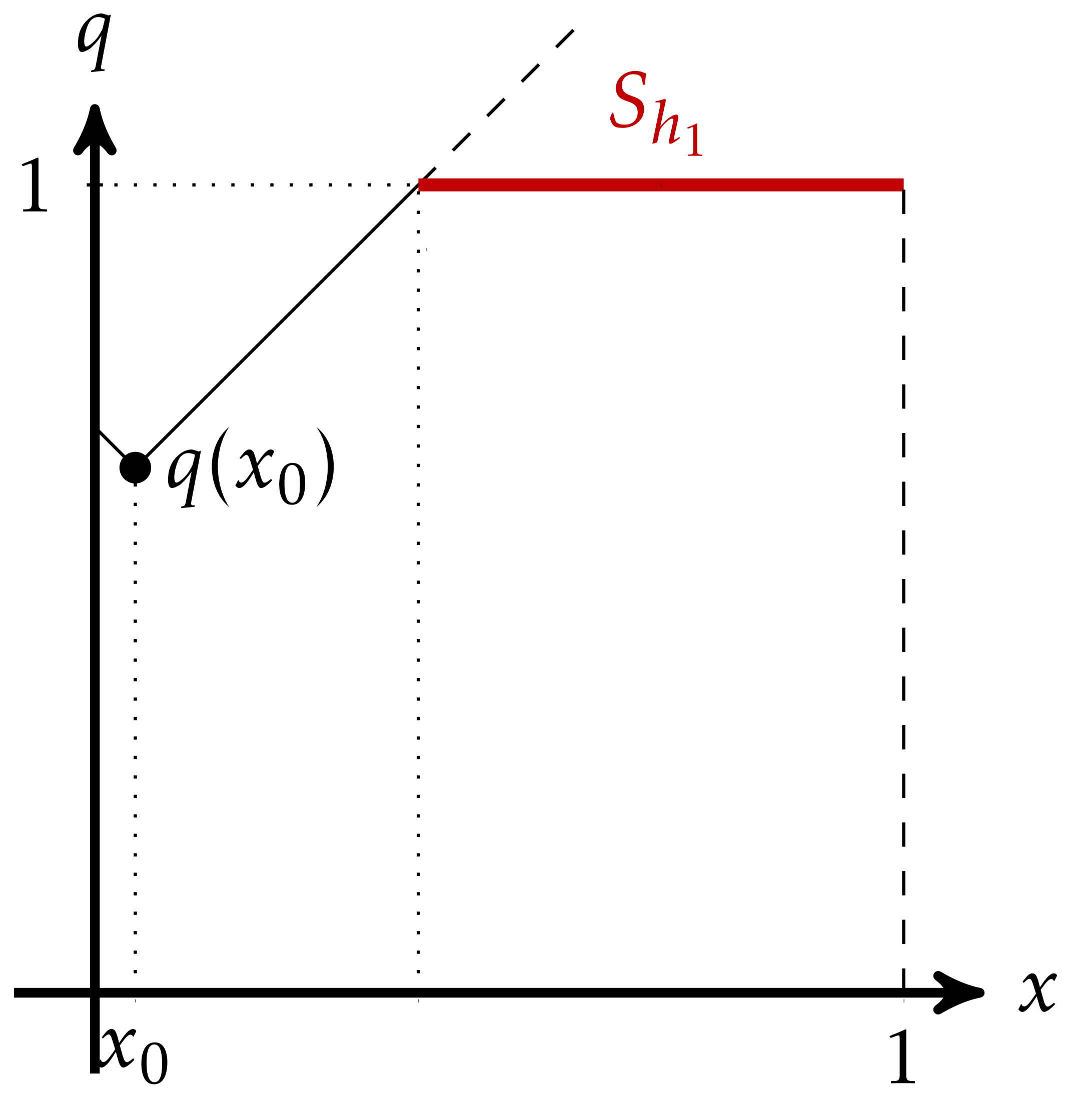
$$\sigma^* \in \operatorname*{argmax}_{\sigma \in \Sigma} \min_{q \in Q} p\left(h_q^\sigma\right)$$

Outline

1. Model
2. Simple search policies
3. Proof ideas
4. Discussion

Search window

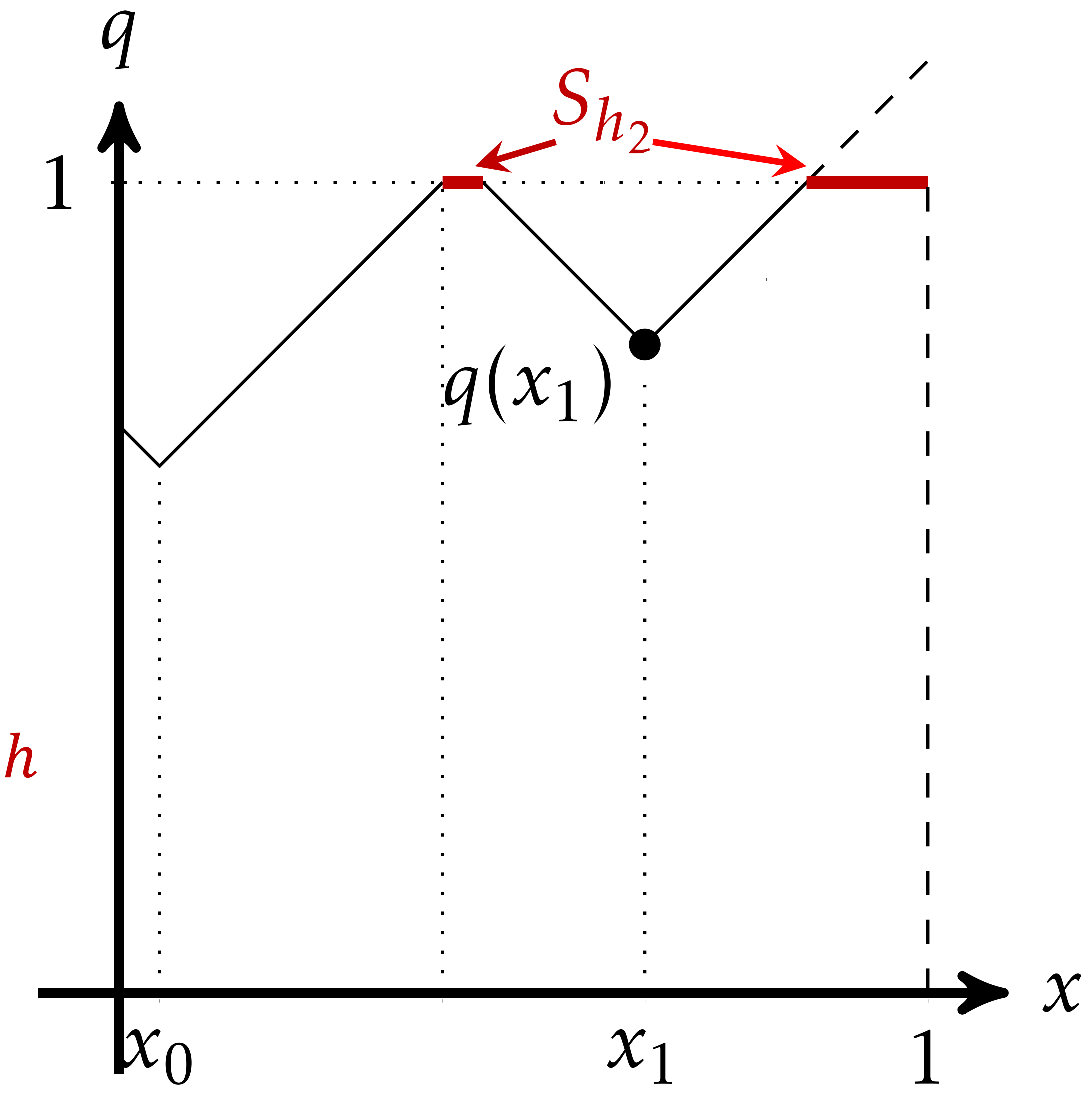
$S_h \equiv \{x \in S | \, \exists q \in Q_h \; s.t. \; q(x) = 1\}$



Search window

$S_h \equiv \{x \in S | \exists q \in Q_h \; s.t. \; q(x) = 1\}$

$\lambda(h)$: Lebesgue measure of search window at h



Policies

For any $\sigma \in \Sigma$, the **reachable histories** are:

$$H^{\sigma} \equiv \{h \in H \mid h \text{ is on path for } \sigma \text{ from } h_0 \text{ for some } q \in Q\}$$

A **policy** is a restriction of $\sigma \in \Sigma$ to H^{σ}

A policy is **optimal** if it can be extended to an optimal strategy

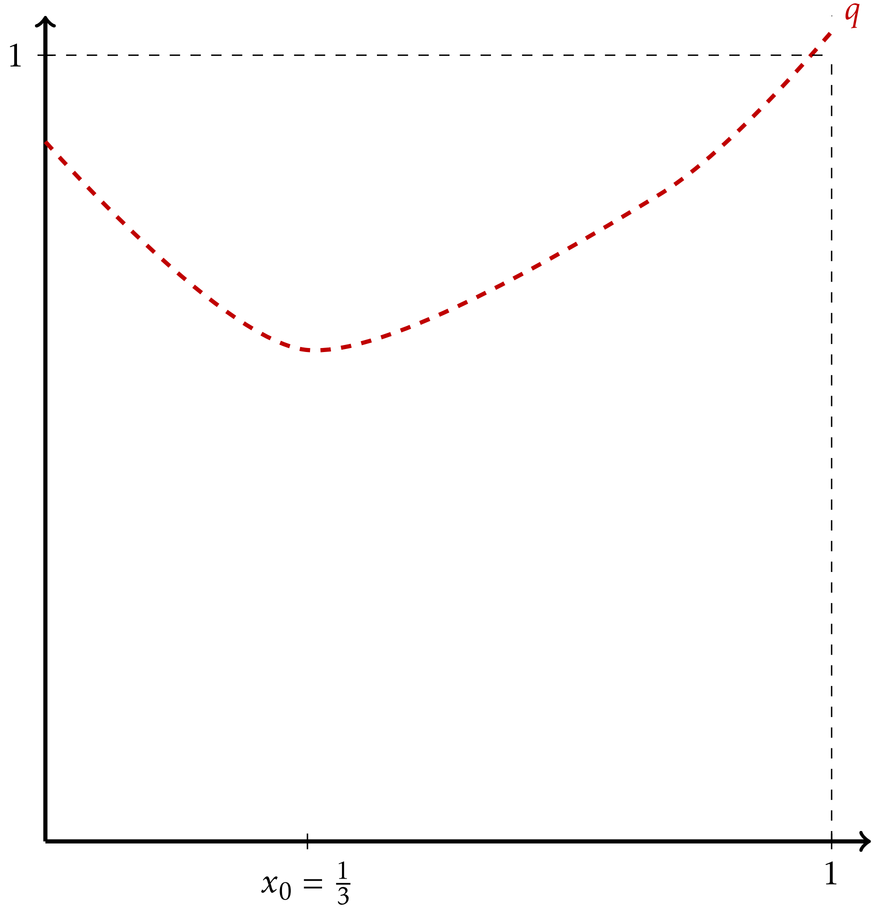
An example search policy, σ

At every history $h \in H^\sigma$

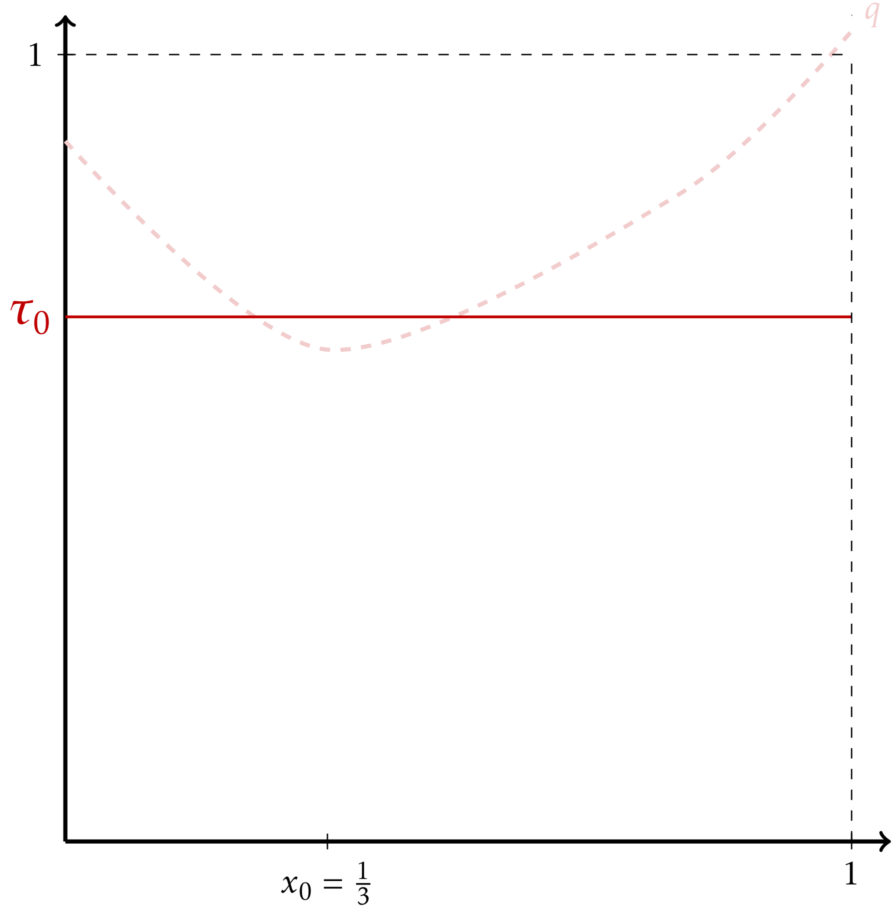
→ stop if $\max_{i \in \{0,\dots,t-1\}} z_i \geq 1 - \frac{1}{3}\lambda(h_t)$

→ search at $1 - \frac{2}{3}\lambda(h_t)$

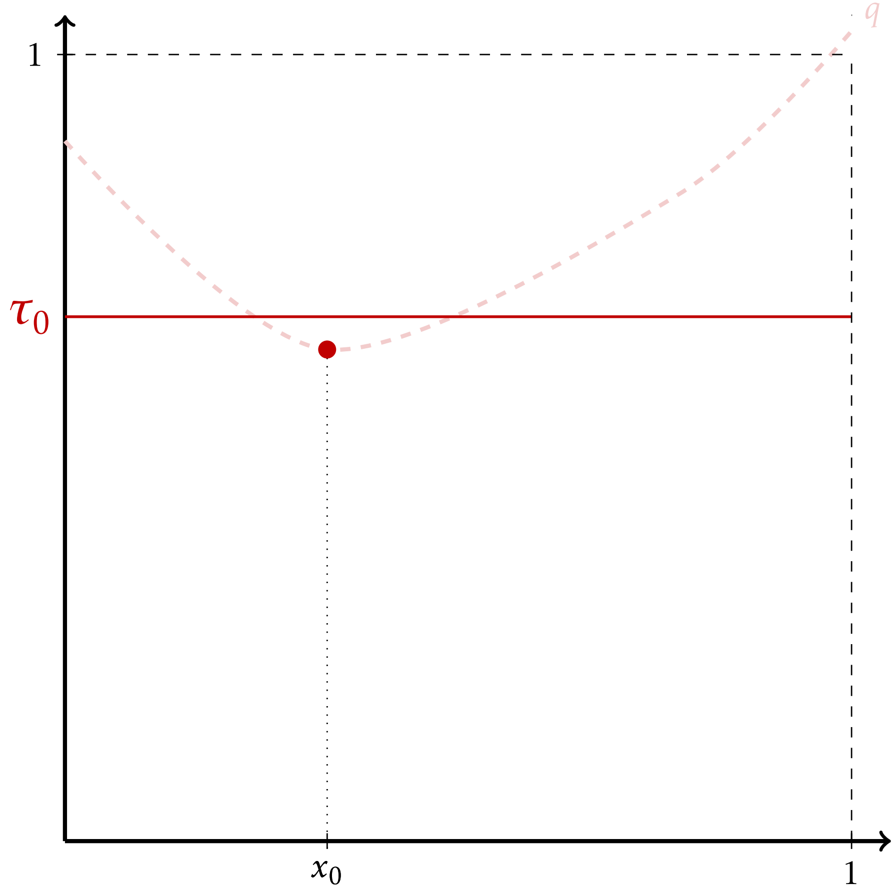
An example search policy



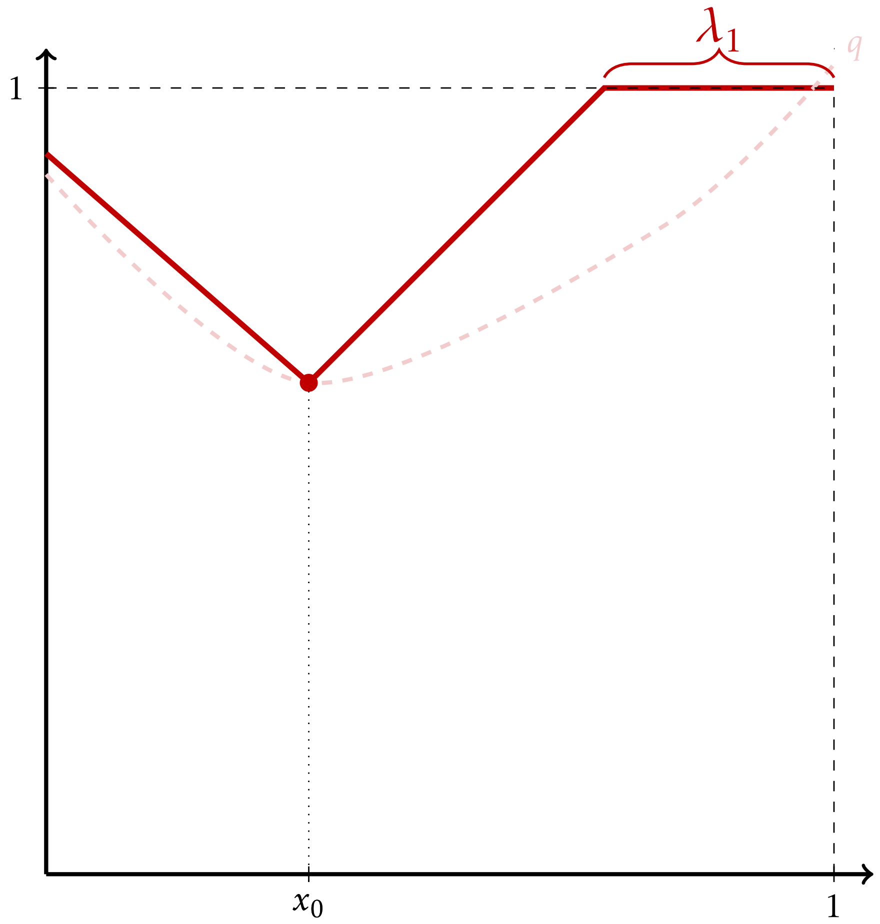
An example search policy



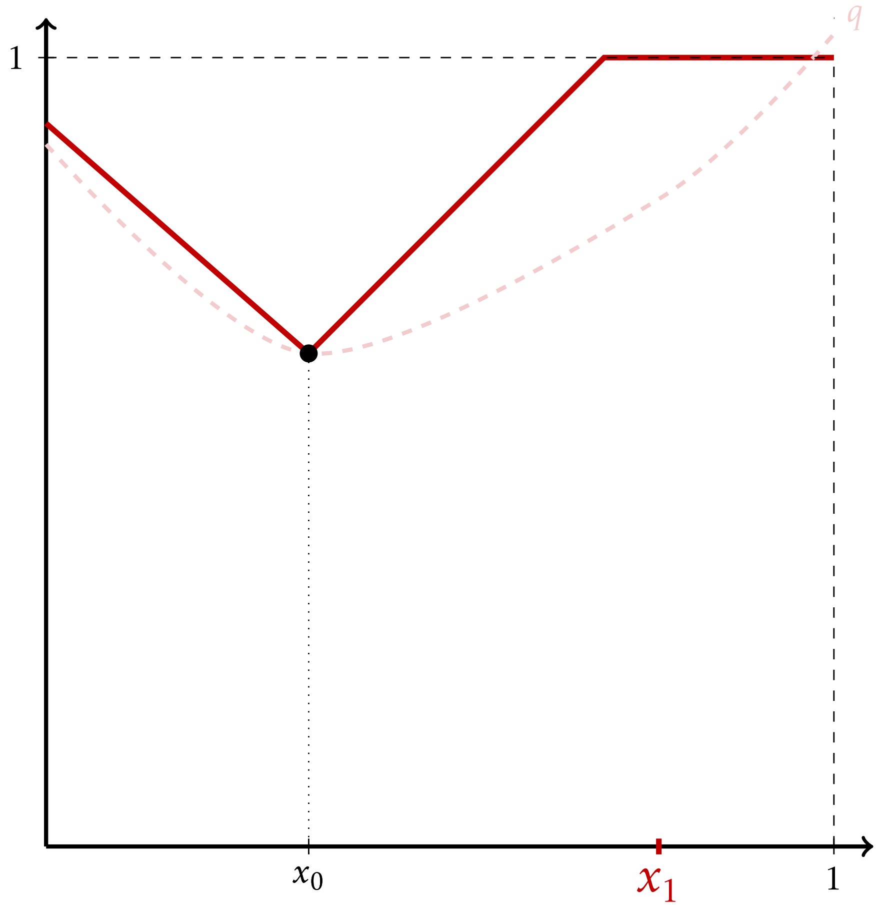
An example search policy



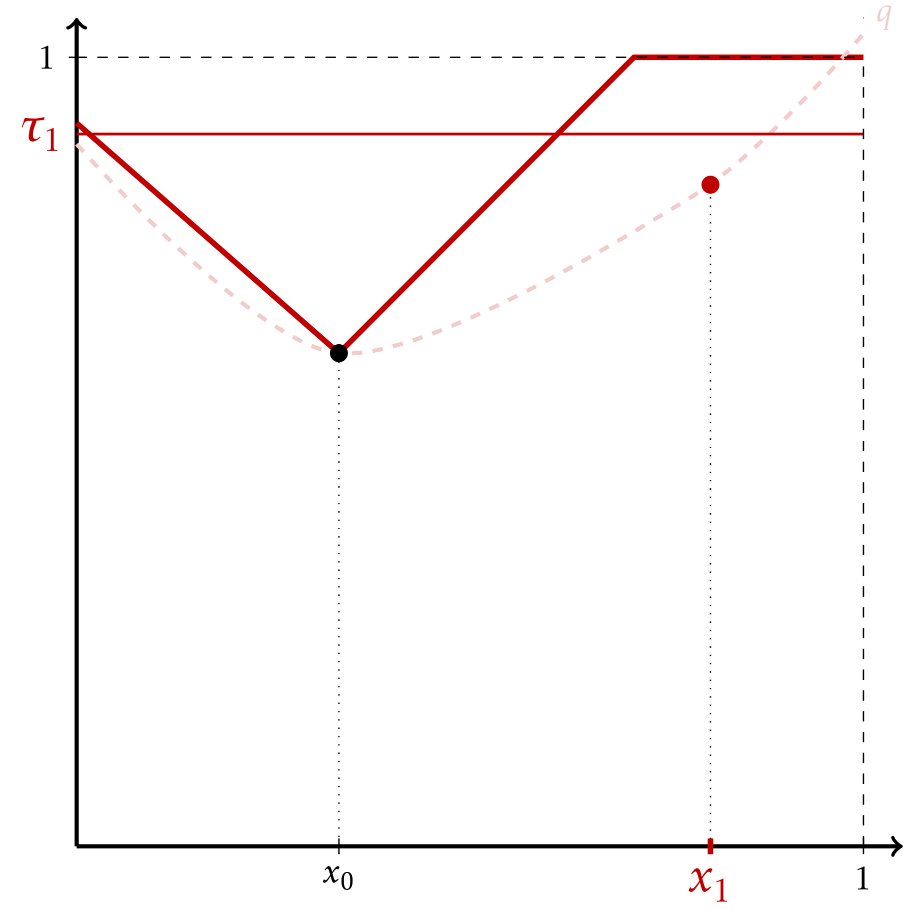
An example search policy



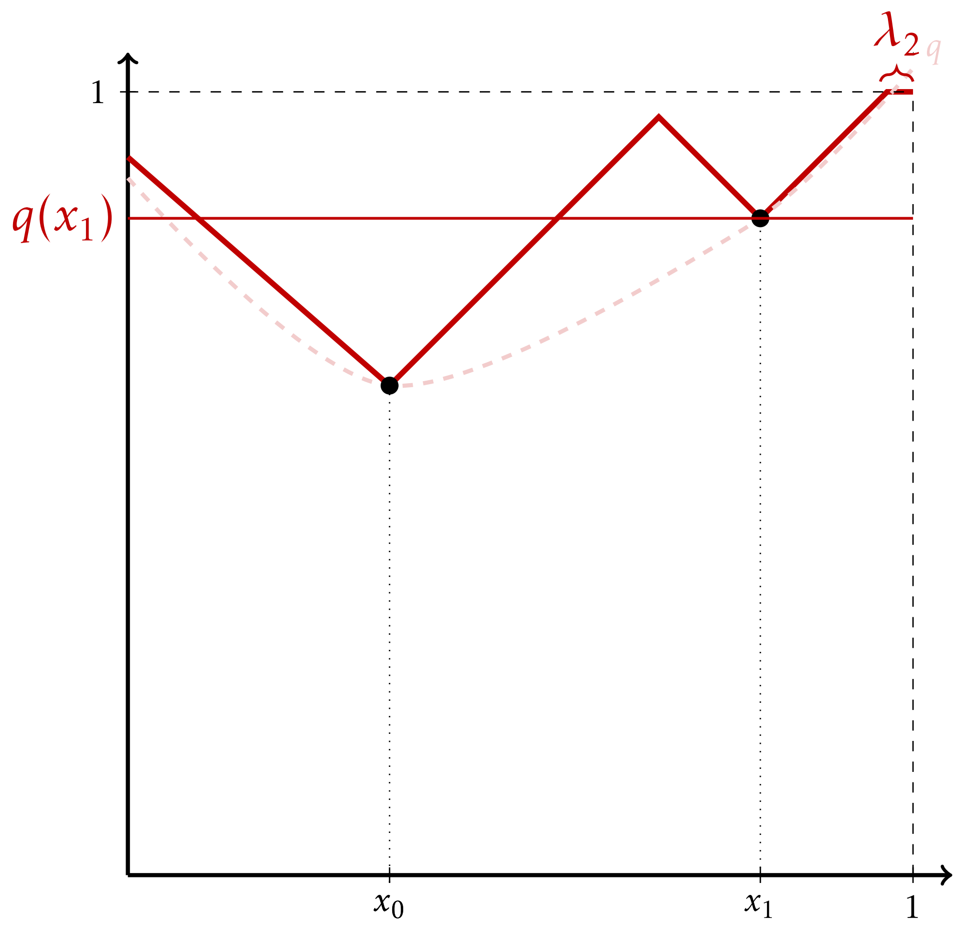
An example search policy



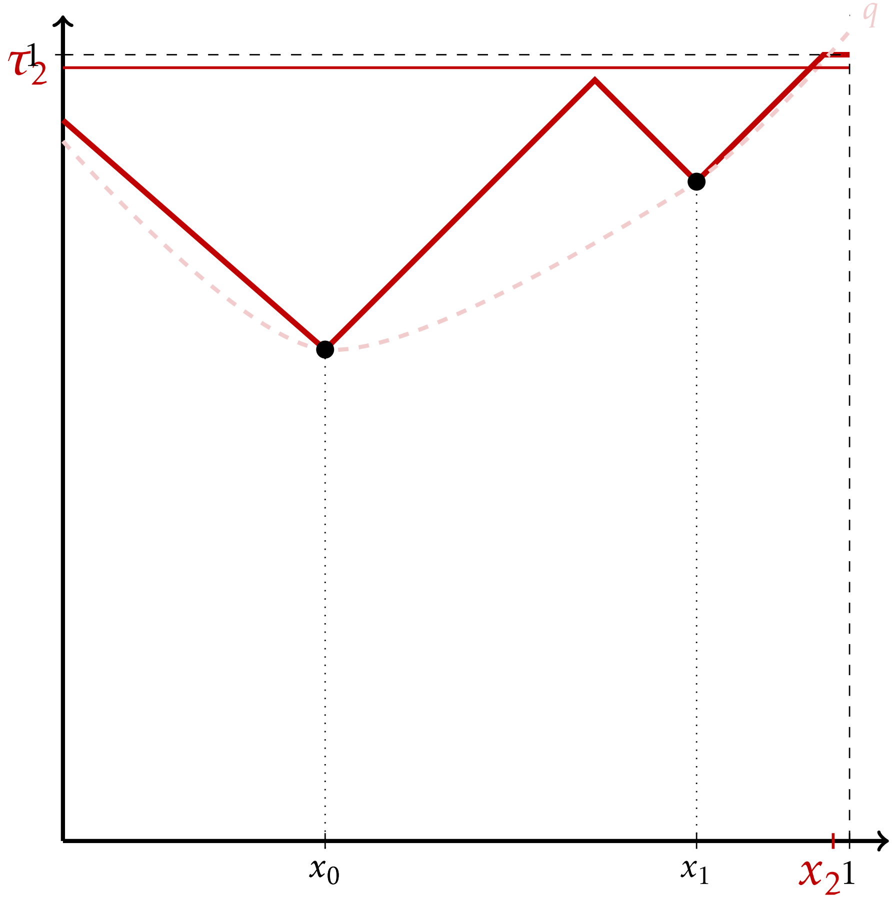
An example search policy



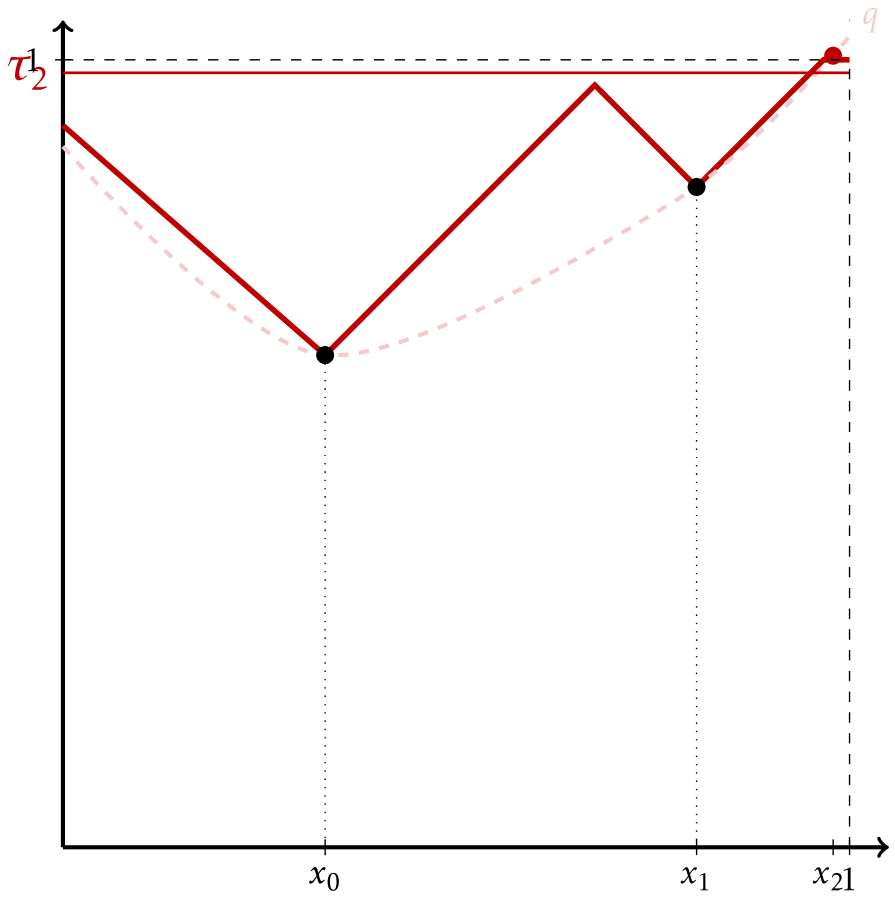
An example search policy



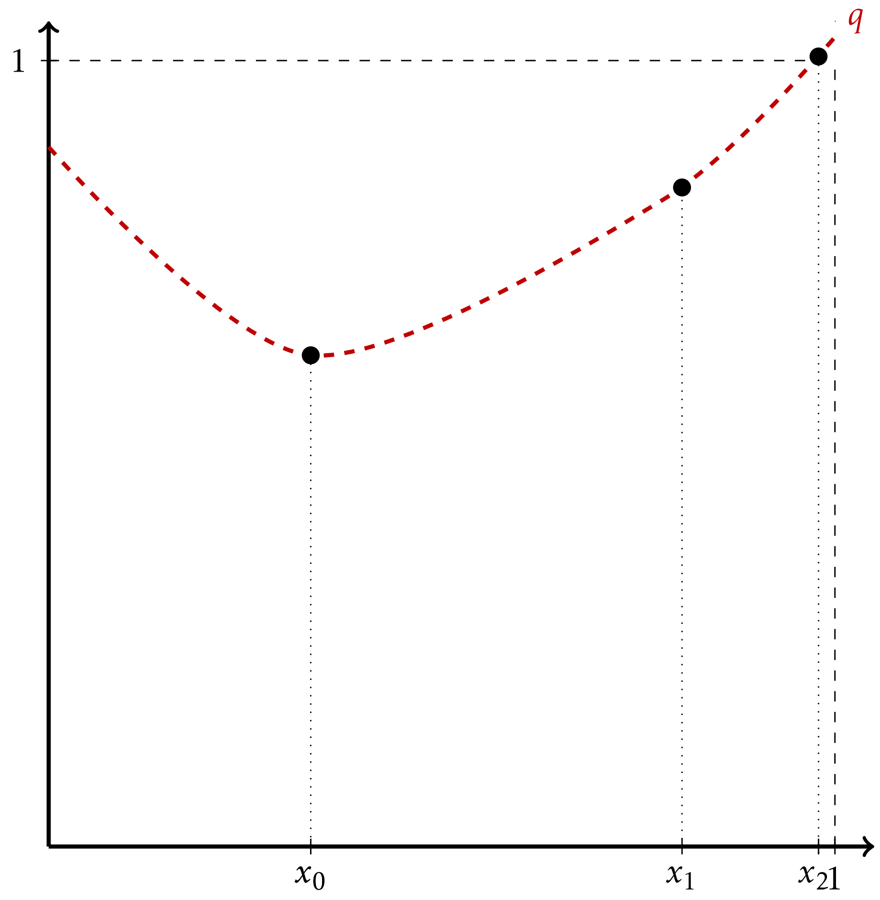
An example search policy



An example search policy



Example search policy



Example search policy

Search policy σ

- explores **incrementally**
- depends only on **search window length**
- stops only if quality exceeds some **threshold**
- **does not invoke recall**

Classes of policies

σ is an **incremental policy** if for every $h \in H^{\sigma}$,

1. either $\sigma(h) = \emptyset$
2. or $\sigma(h) > x$ for all x explored in h

Classes of policies

σ is an **index policy** if there exists $\hat{\sigma}: \mathbb{R}_+ \to [0, 1] \cup \{\emptyset\}$ such that $\sigma(h) = \hat{\sigma}\big(\lambda(h)\big)$ for every non-terminal history $h \in H^\sigma$.

Classes of policies

σ is a **threshold policy** if for every non-terminal $h_t \in H^\sigma$, there exists τ_{h_t} such that for every $h_{t+1} \in H^\sigma$ that follows it, $\sigma(h_{t+1}) = \emptyset$ if and only if $z^*_{h_{t+1}} \geq \tau_{h_t}$

Classes of policies

σ **does not invoke recall** if the searcher always takes the last item discovered

Solution concept revisited

σ^* is a **dynamically consistent policy** if at all histories $h \in H^{\sigma^*}$:

$$\sigma^* \in \operatorname*{argmax}_{\sigma \in \Sigma} \min_{q \in Q_h} p\left(h_q^\sigma\right)$$

Main result

Theorem: There exists an optimal policy that is incremental, threshold, index, does not invoke recall, and is dynamically consistent.

Rediscovery is a simple process

1. Searcher can explore the space freely
   Suffices to **search in order**
2. Searcher can use complex stopping rules
   Threshold rules suffice
3. Searcher has perfect recall
   Recall is never invoked
4. Histories are complex
   Search window length is **only state variable**

Outline

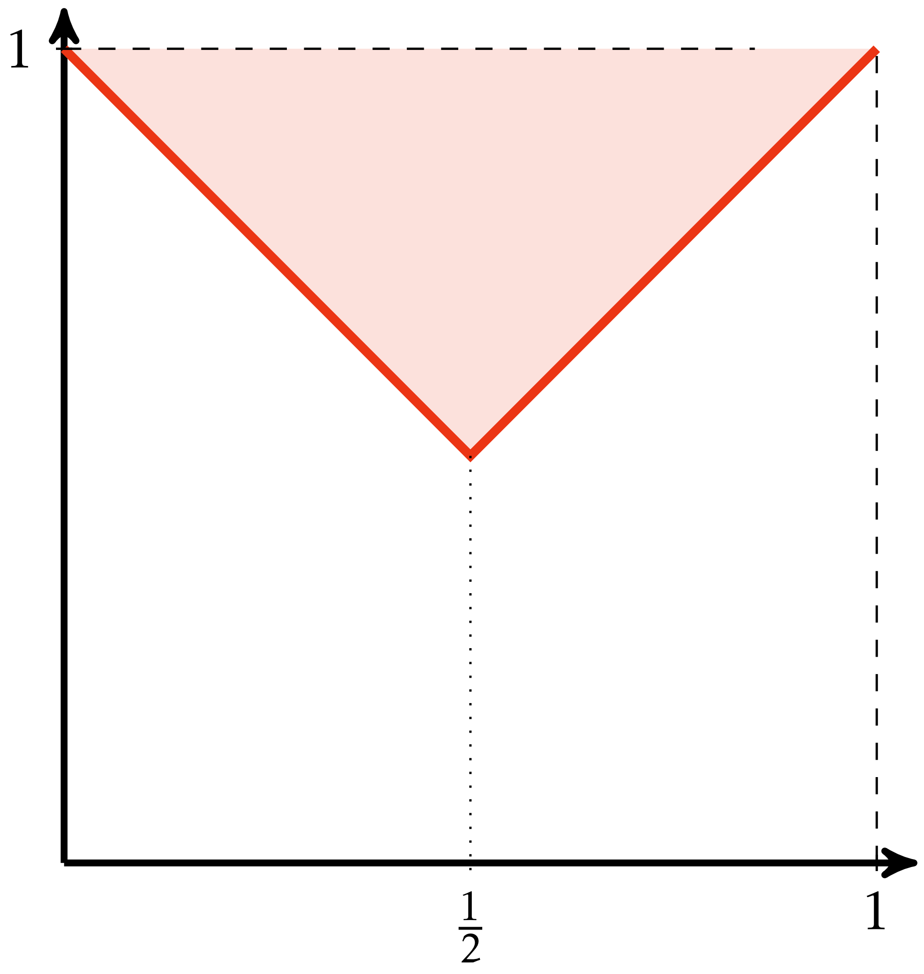
1. Model
2. Simple search policies
3. Proof ideas
4. Discussion

Warm-up: two period case

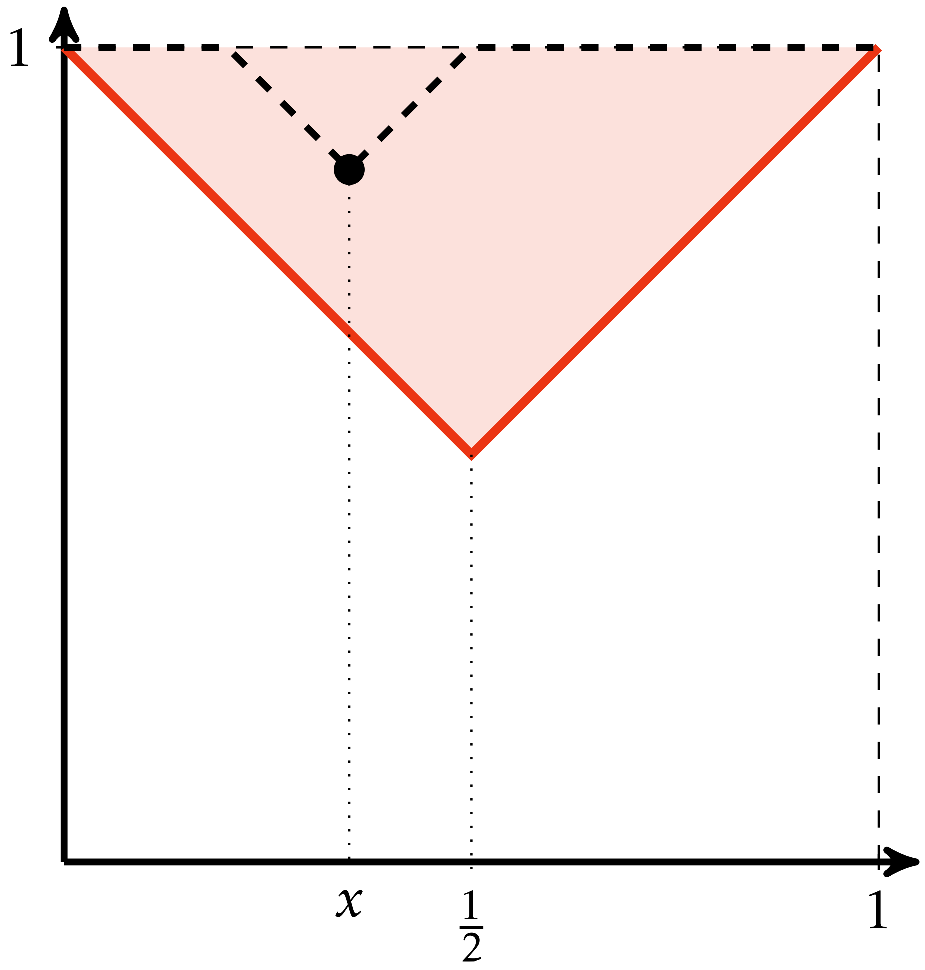
Assumption: suppose search must end in two periods

Searcher's optimal strategy hedges against two 'risks'

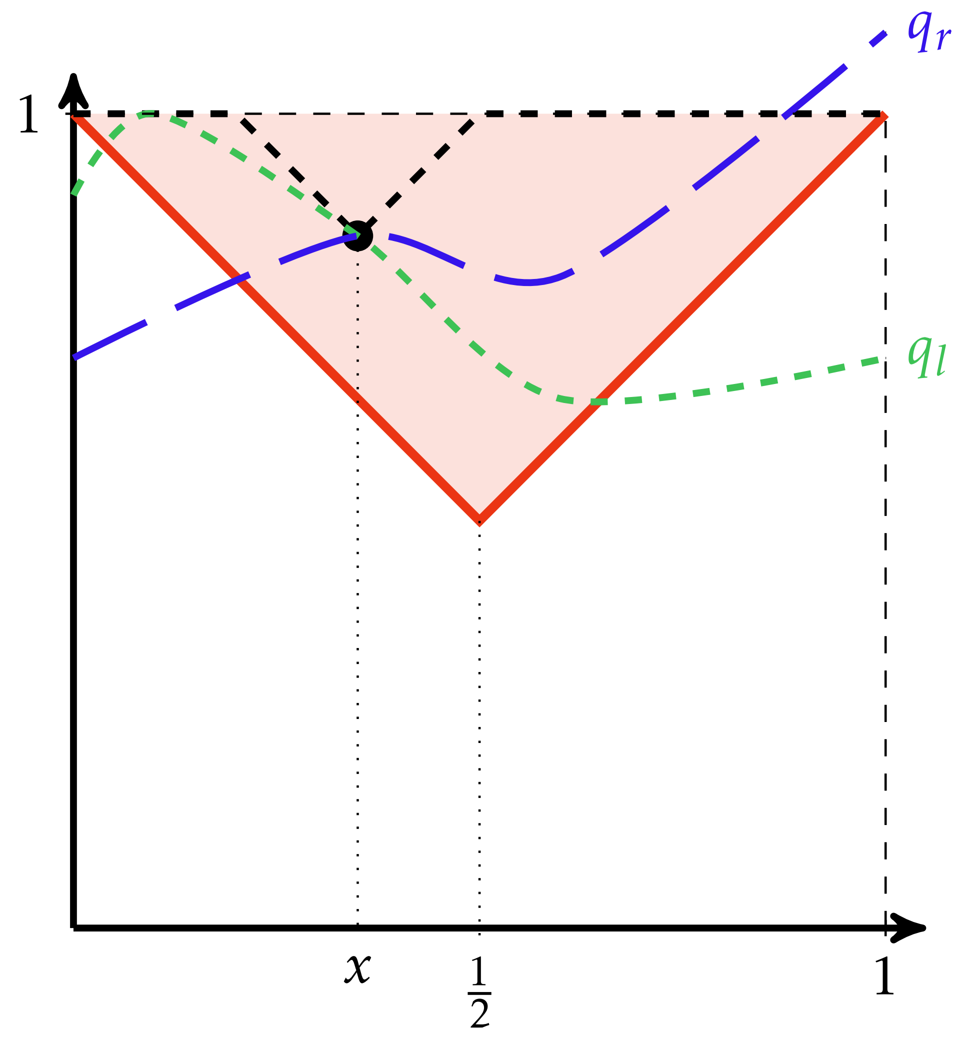
Bifurcation risk



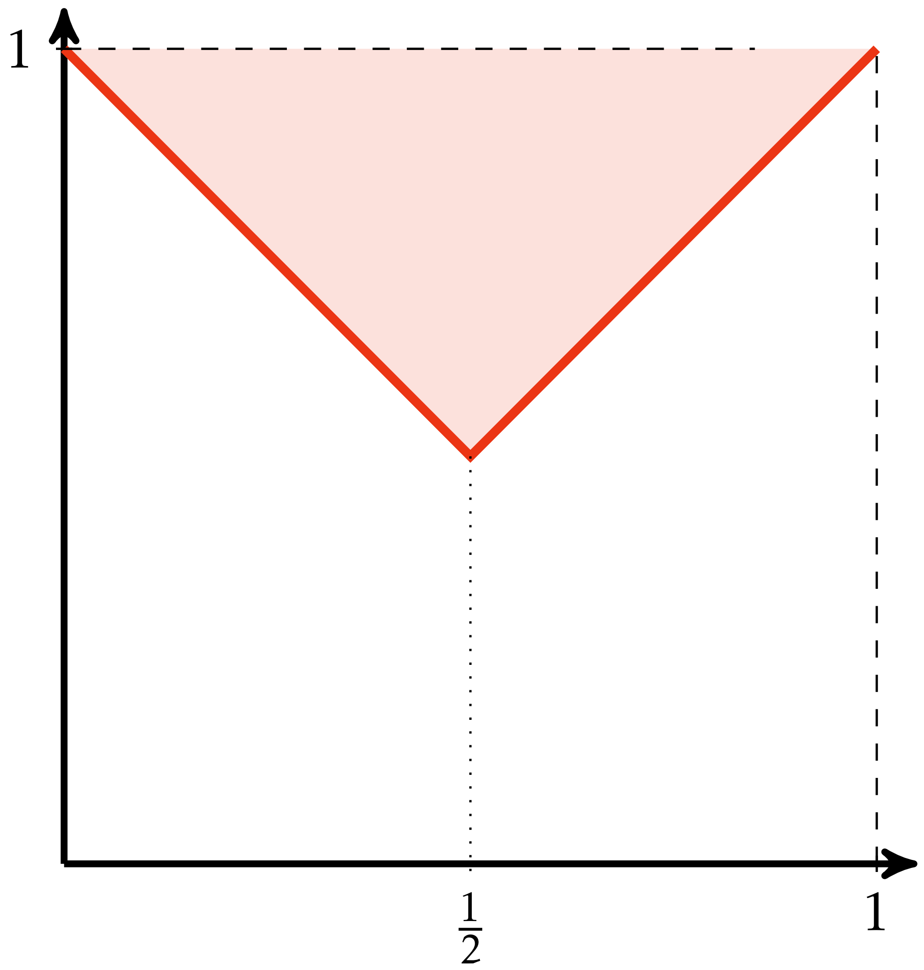
Bifurcation risk



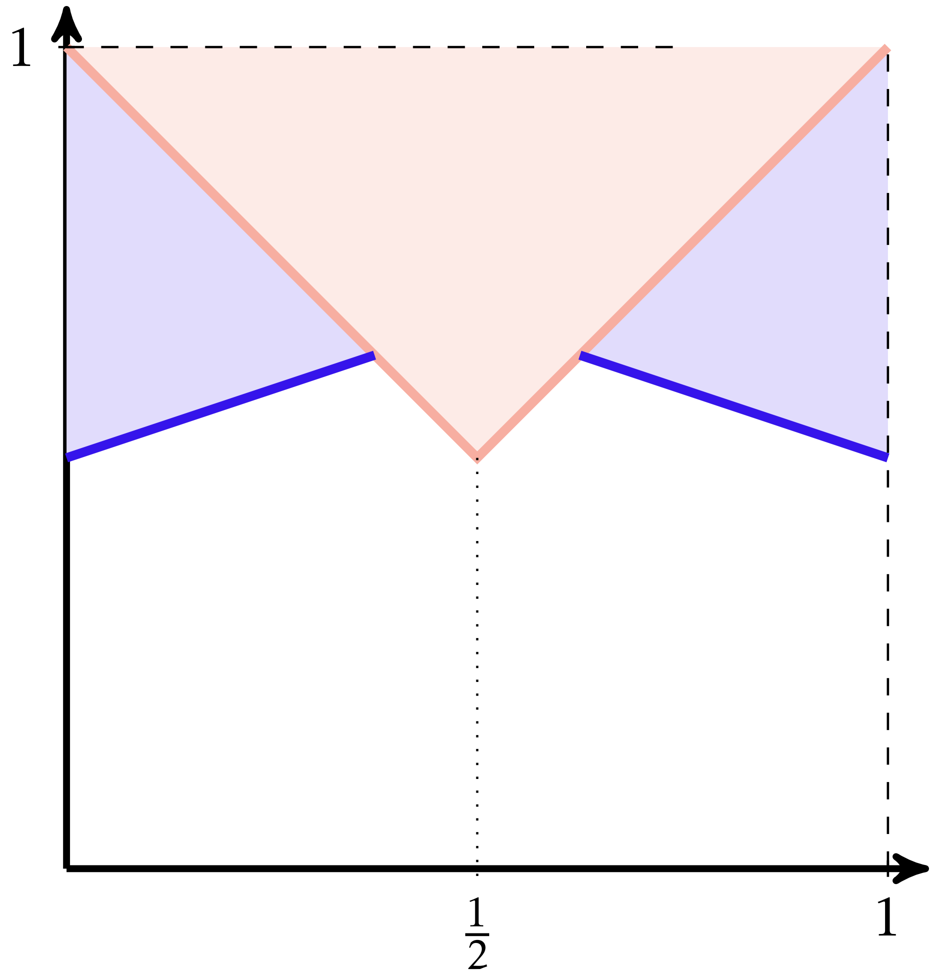
Bifurcation risk



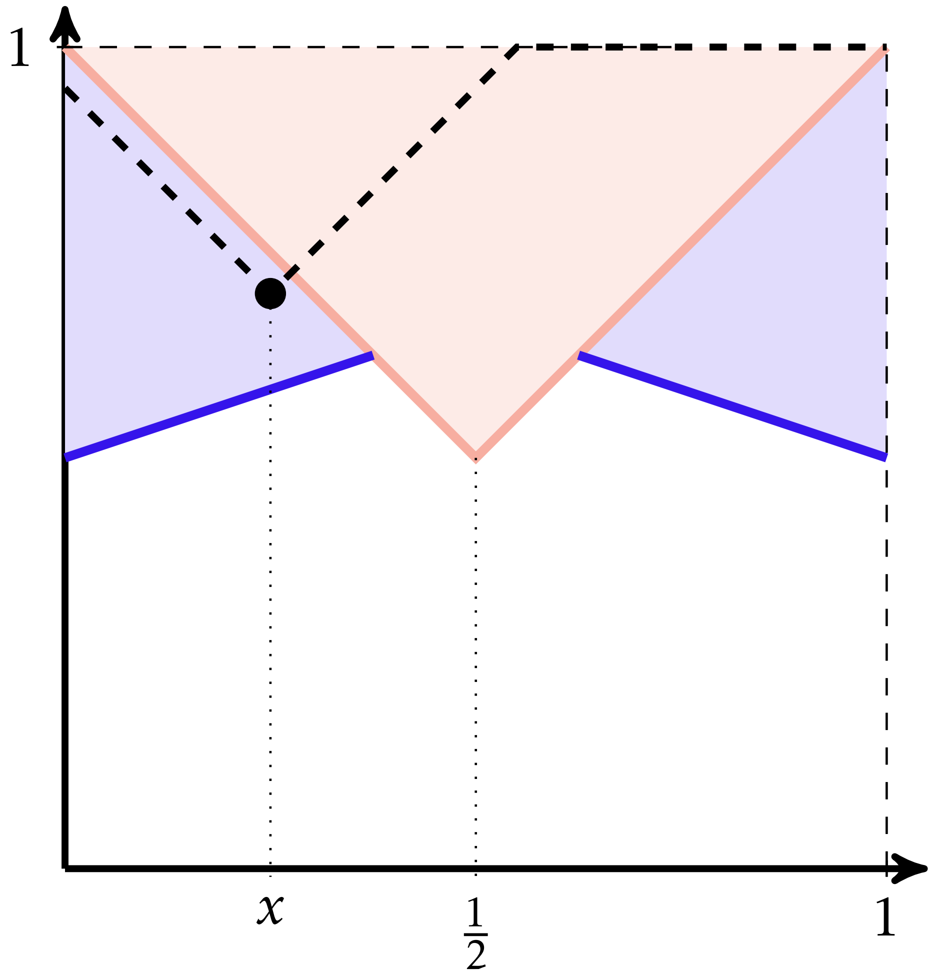
Bifurcation risk



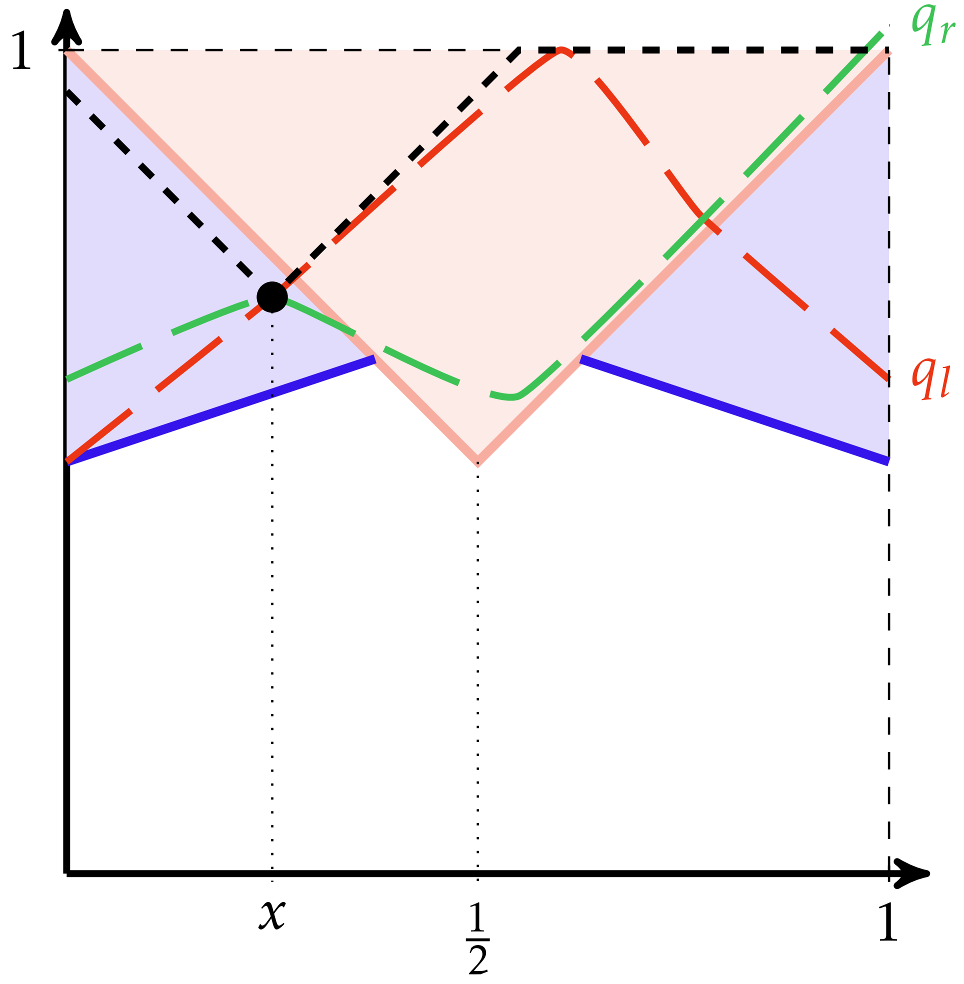
Directional risk



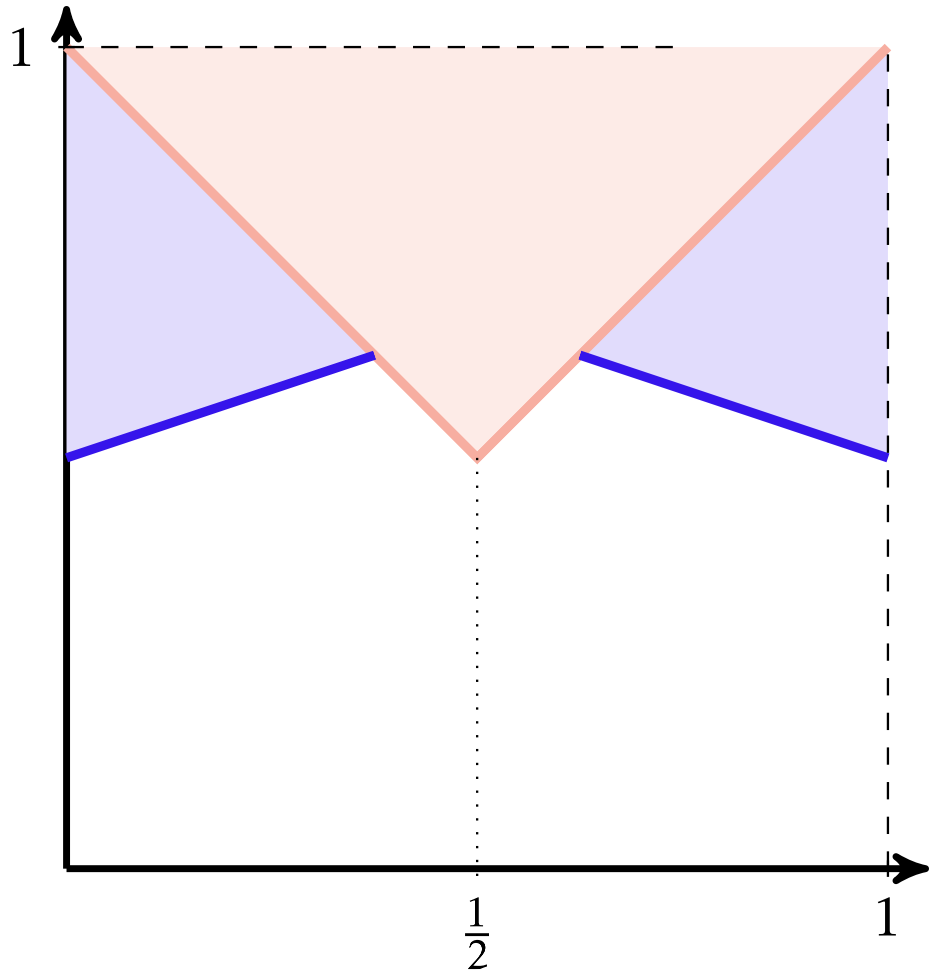
Directional risk



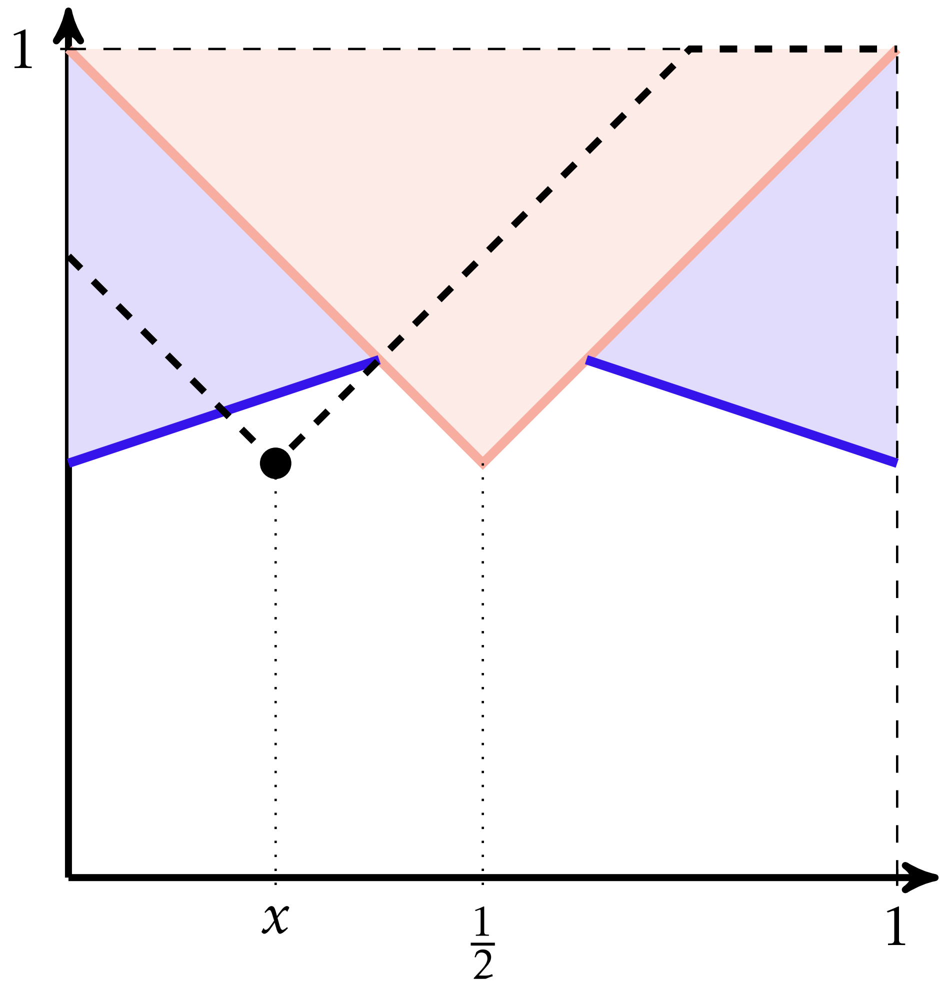
Directional risk



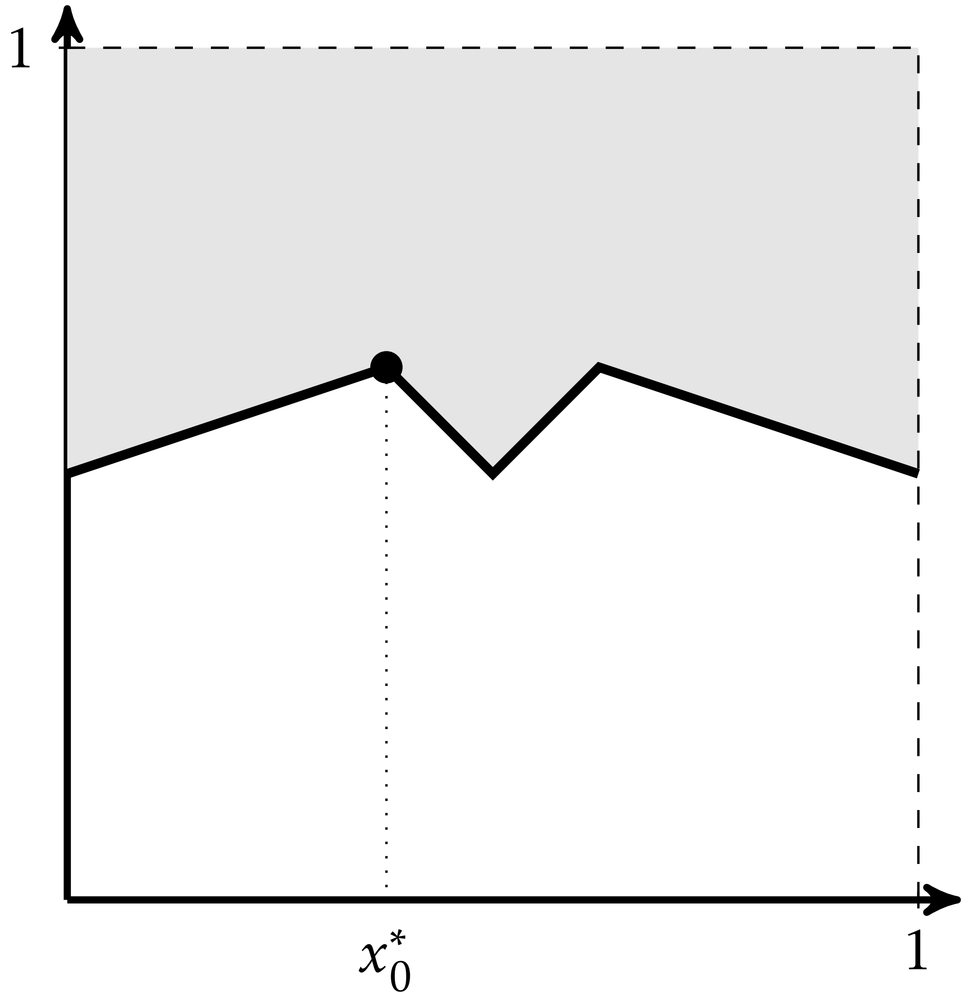
Directional risk



Continuation region



Optimal search policy



Theorem

Search intensity: $N: [0,1]^2 \to \mathbb{N}$

$$N(c,l) = \begin{cases} 0, & c \in (1-\frac{l}{2}, 1] \\ 1, & c \in (\frac{l}{2}, 1-\frac{l}{2}] \\ n, & c \in (\frac{l}{n(n+1)}, \frac{l}{n(n-1)}] \end{cases}$$

Threshold: $\tau: [0,1]^2 \to \mathbb{R}$

$$\tau(c,l) = 1 - \frac{l}{2N(c,l)} - \frac{N(c,l)-1}{2}c$$

Optimal policy:

If $S_h = [a,1]$ and $N\big(c,\lambda(h)\big) > 0$:

$$\sigma^*(h) = 2 - \lambda(h) - \tau\big(c,\lambda(h)\big)$$

If $S_h = [a,b] \sqcup [c,1]$ or $N\big(c,\lambda(h)\big) = 0$:

$$\sigma^*(h) = \emptyset$$

Outline

1. Model
2. Simple search policies
3. Proof ideas
4. Discussion

Features of optimal search

1. **Process of elimination**:

   Bad discovery→ remove similar choices from consideration

2. **Satisficing**

   "[so] long as the problem is not solved, search will continue." (Cyert and March 1963)

3. **Increasing thresholds**

   Emboldened by failure **because objective is attainable.**

Roger Bannister and the 4min mile

4 min mile seriously attempted since 1880s

Physical or psychological barrier?

Roger Bannister—iconoclast, lone-wolf, no coaches, own system

Bannister breaks 4min time first in 1954 in bad conditions

Many others soon follow, and now is not so rare a feat

"…what goes for runners goes for leaders running organizations… Whether it's an executive, an entrepreneur, or a technologist, some innovator changes the game, and that which was thought to be unreachable becomes a benchmark, something for others to shoot for. That's Roger Bannister's true legacy… "

- Bill Taylor, *Harvard Business Review*

Summary

Knowing something is discoverable affects how you search for it

Optimal search is a **process of elimination**

Foundation for behavioral theories of firm search and R&D

Literature

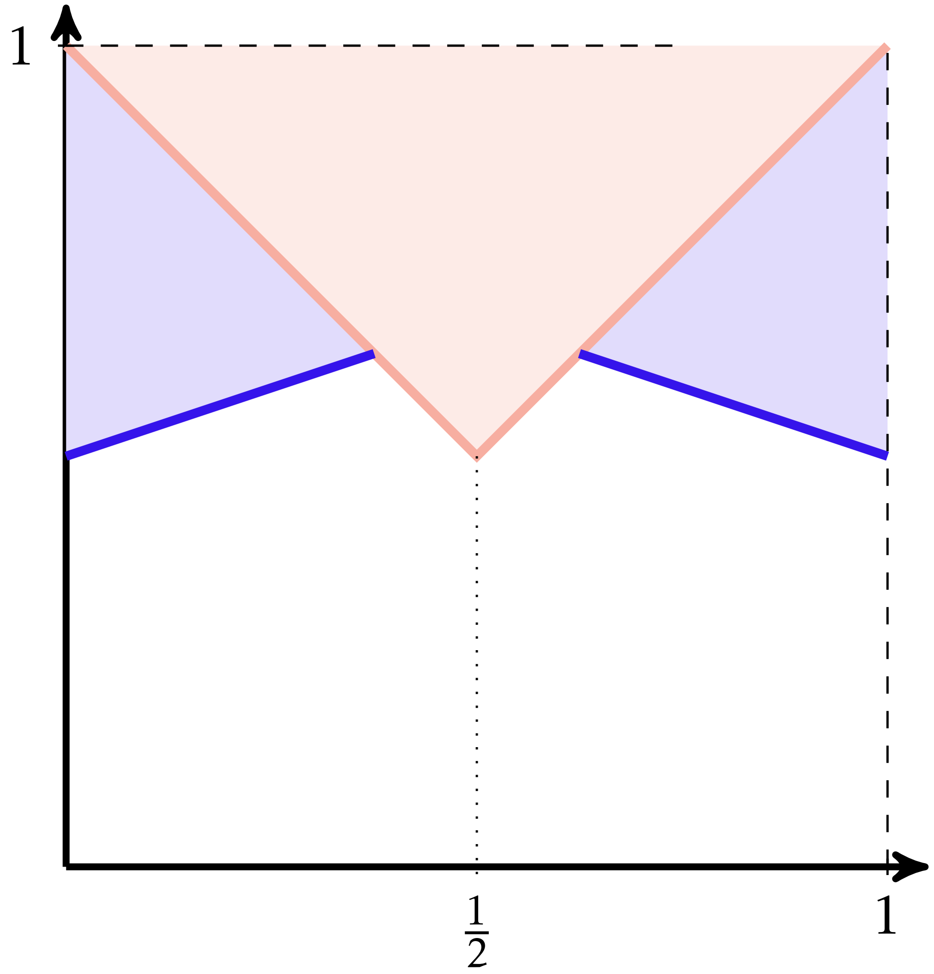
Search: Weitzman (1979), Rothschild (1978), Callander (2011), Malladi (2022)…

Multi-arm bandits and optimization: Slivkins (2019), Hansen et al (1992), Radner (1975), Francetich and Kreps (2015)…

Problemistic Search and Rugged Landscapes: Cyert and March (1963), Levinthal (1997), Bellinger et al. (2012), Garfagnini and Strulovici (2016), Callander, Lambert and Matouschek (2022)…

Dynamics and ambiguity: Klibanoff and Hanany (2009),…

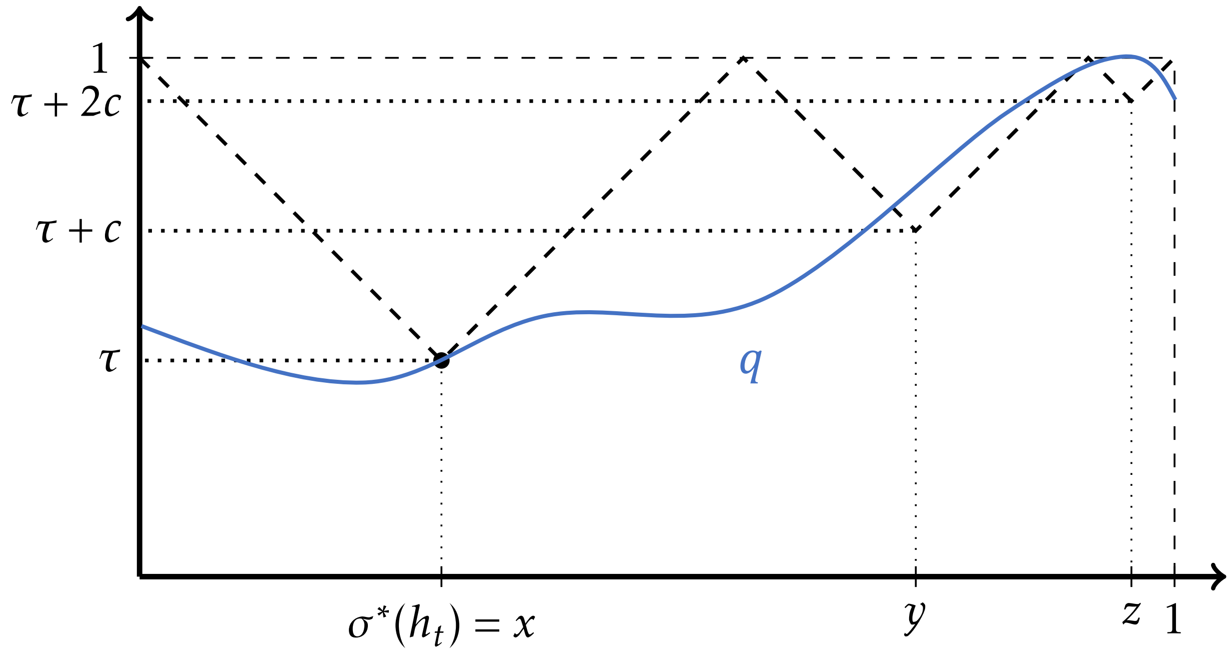
Thank you!



Proof steps

Step 0: guess worst-case $q \in Q_h$ for σ^* when $S_h = [a, 1]$

Step 0: discover τ at $\sigma^*(h_t)$ in worst-case

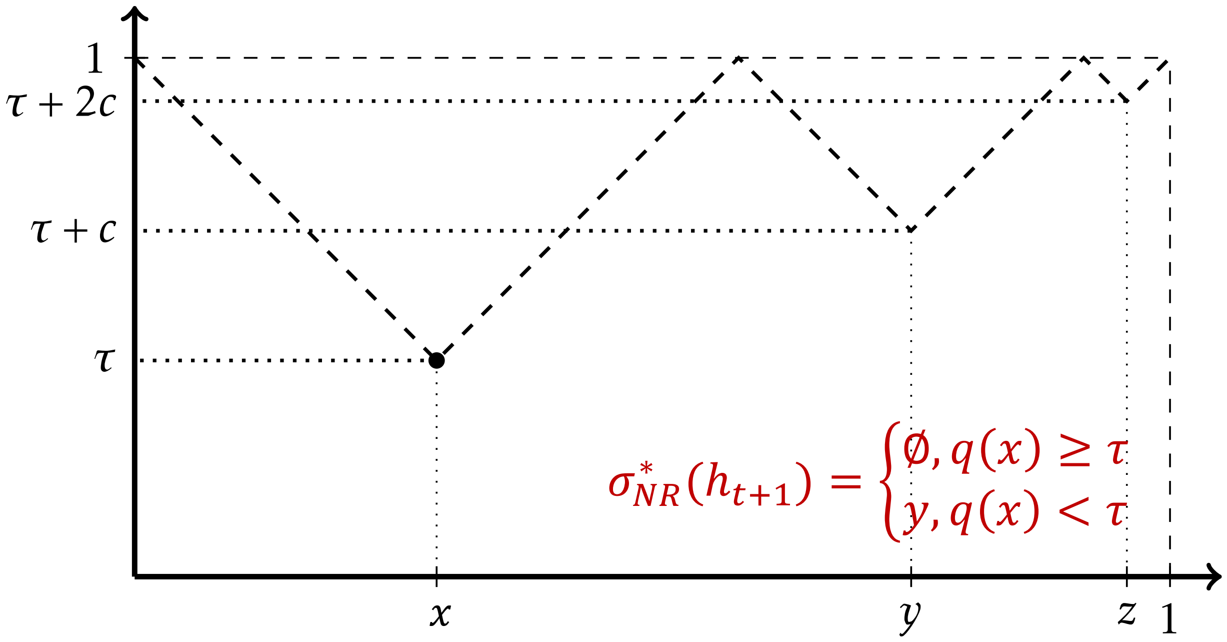


Proof steps

Step 0: guess worst-case $q \in Q_h$ for σ^* when $S_h = [a, 1]$

Step 1: verify this guess, i.e., for any $q' \in Q_h$, $p\left(h_q^{\sigma^*}\right) \leq p\left(h_{q'}^{\sigma^*}\right)$

Step 1a: define σ^*_{NR}

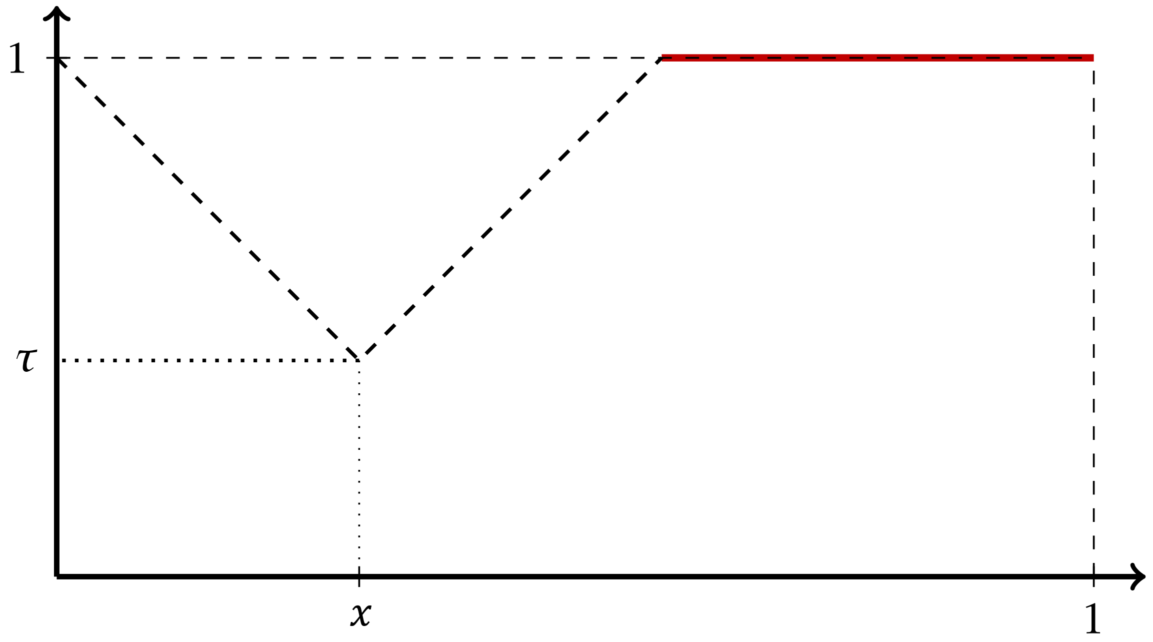


$$\sigma^*_{NR}(h_{t+1}) = \begin{cases} \emptyset, q(x) \geq \tau \\ y, q(x) < \tau \end{cases}$$

$$\sigma^*_{NR}(h_t) = x$$

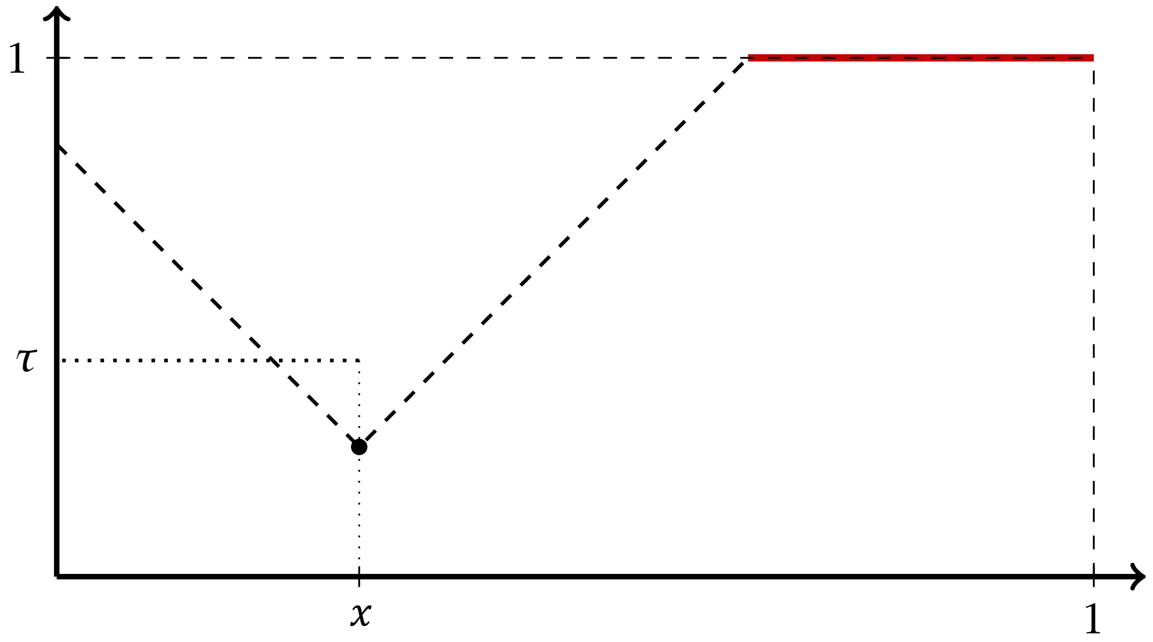
$$\sigma^*_{NR}(h_{t+2}) = \begin{cases} \emptyset, q(y) \geq \tau + c \\ z, q(y) < \tau + c \end{cases}$$

Step 1a: searcher gets at least $\tau - c$ with σ^*_{NR}



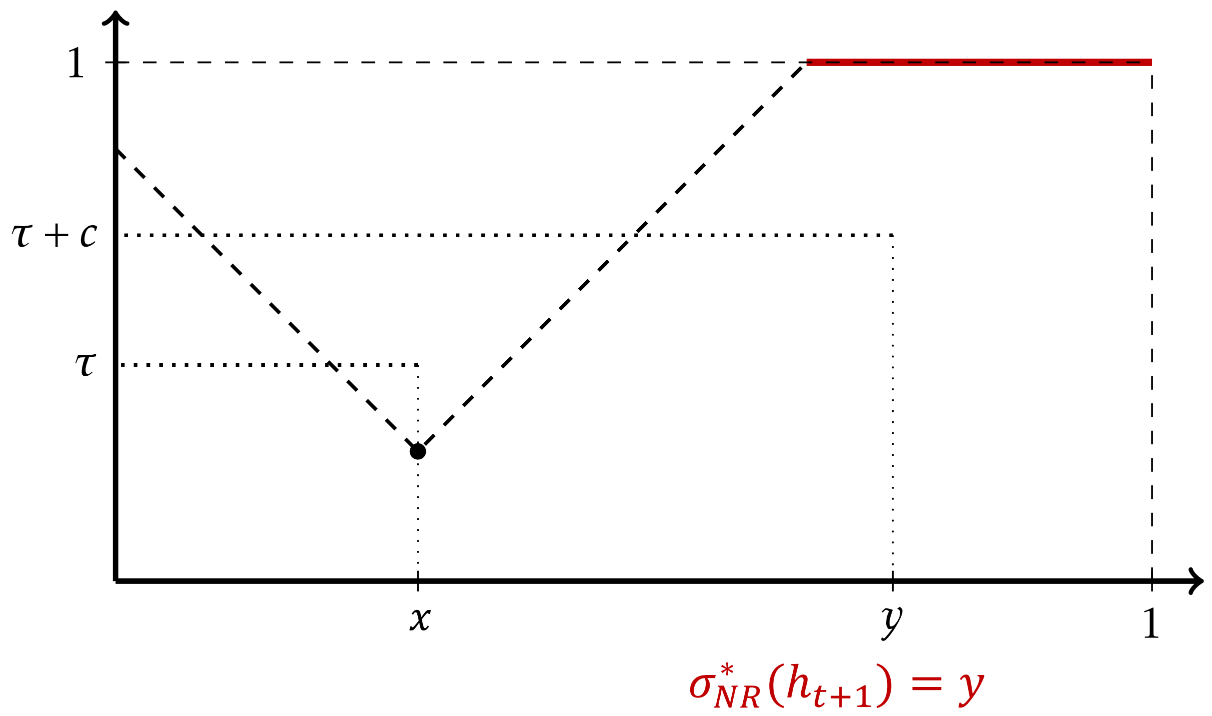
$\sigma^*_{NR}(h_t) = x$

Step 1a: searcher gets at least $\tau - c$ with σ_{NR}^*

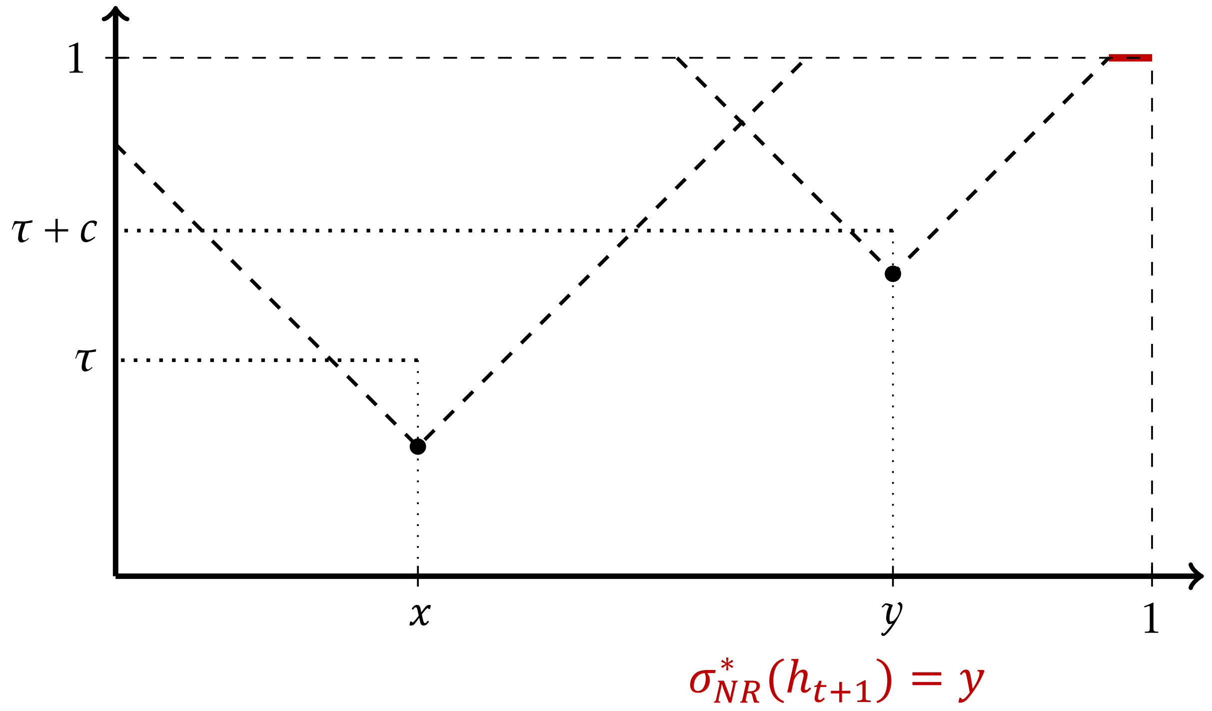


$\sigma_{NR}^*(h_t) = x$

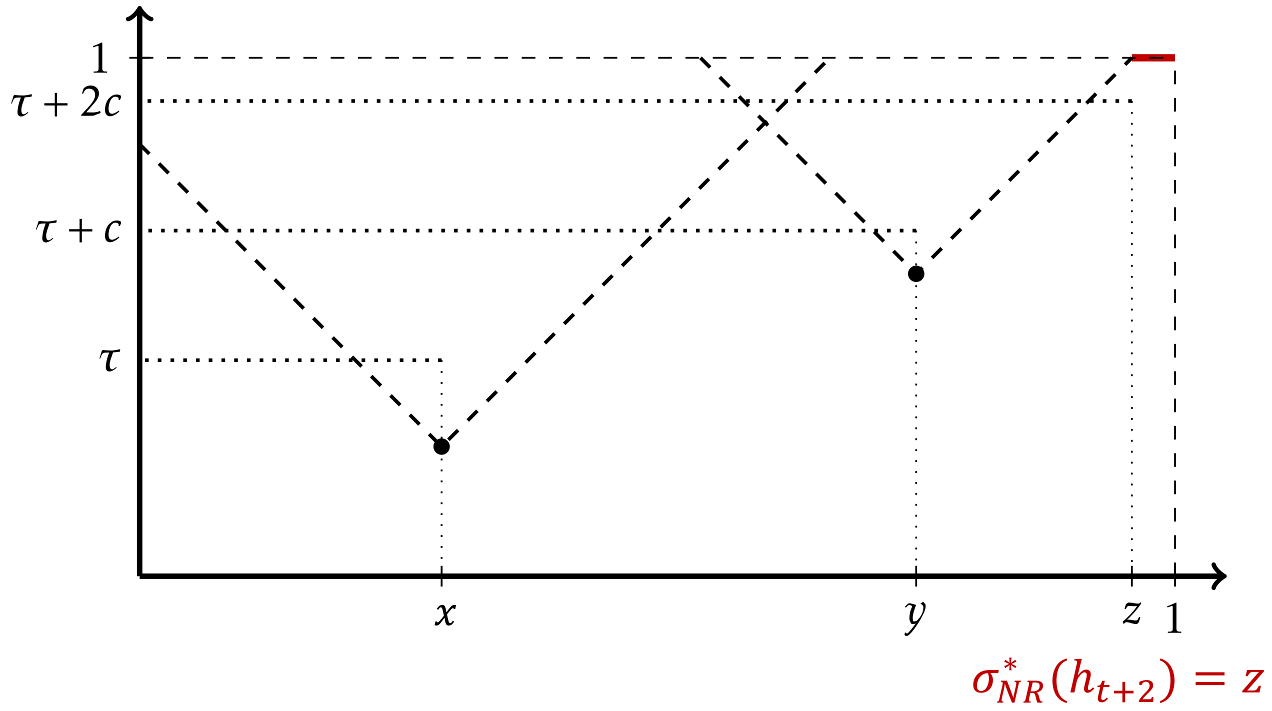
Step 1a: searcher gets at least $\tau - c$ with σ^*_{NR}



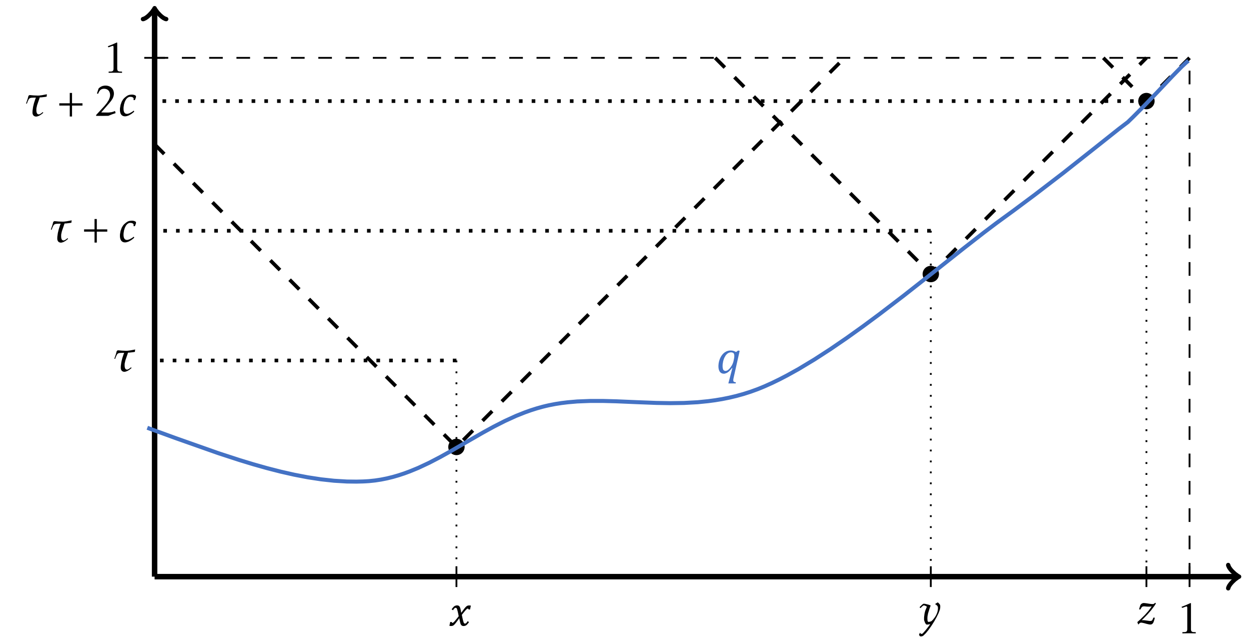
Step 1a: searcher gets at least $\tau - c$ with σ_{NR}^*



Step 1a: searcher gets at least $\tau - c$ with σ^*_{NR}



Step 1a: searcher gets at least $\tau - c$ with σ^*_{NR}

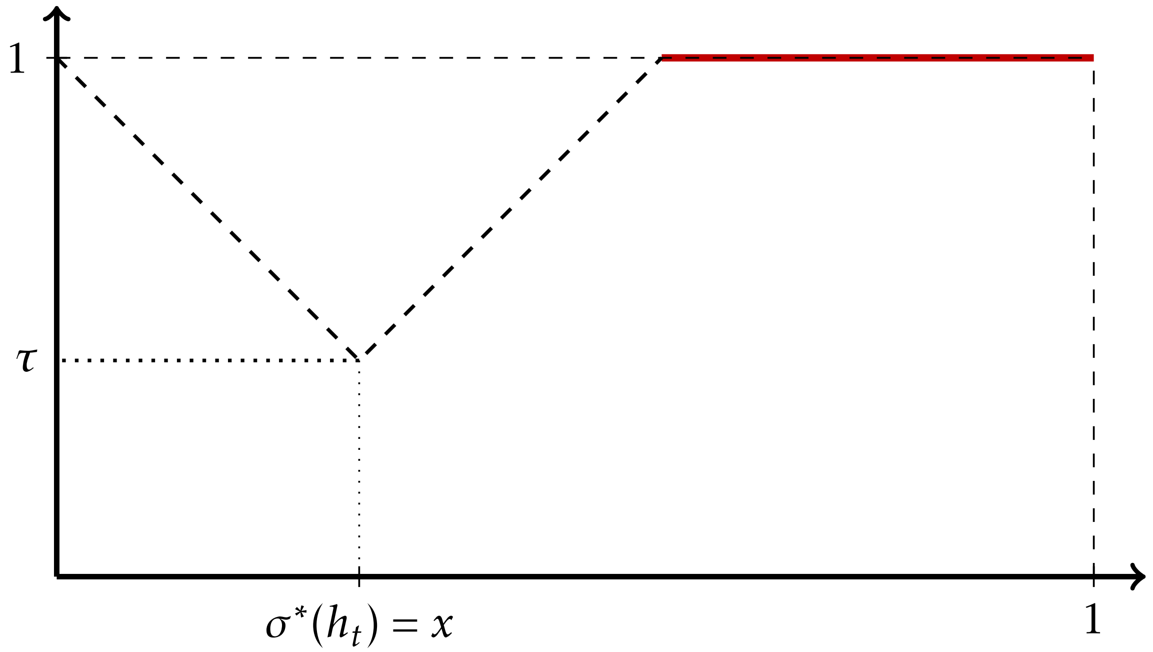


$\sigma^*_{NR}(h_{t+2}) = z$

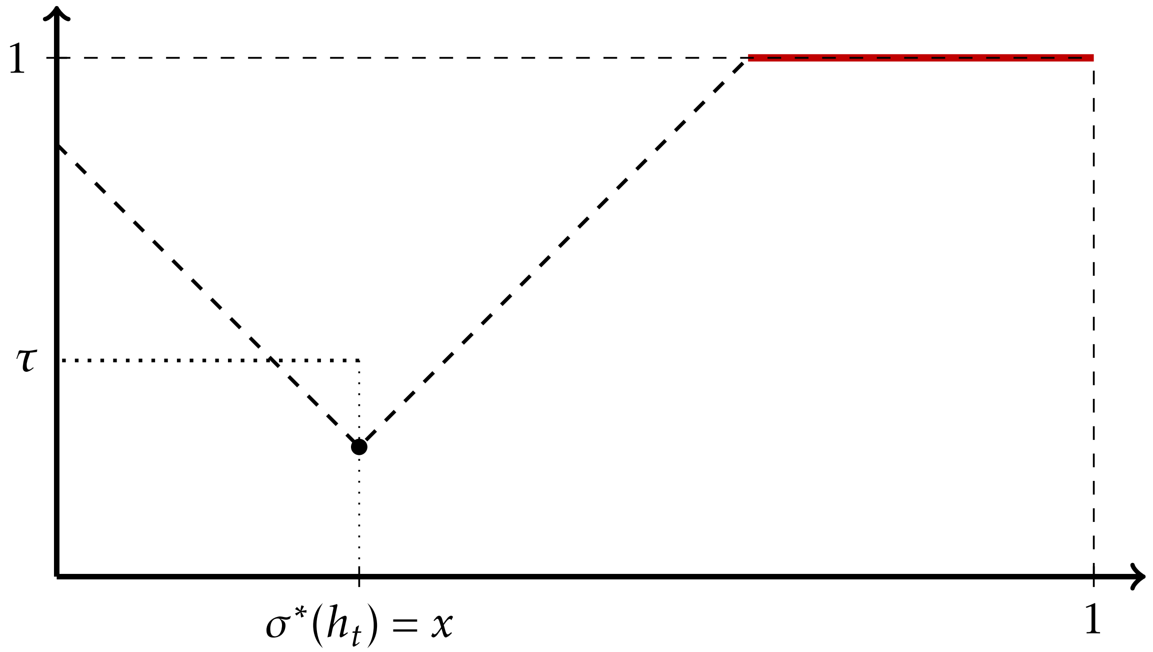
Payoff:

$\tau + 2c \; - 3c = \tau - c$

Step 1: searcher gets at least $\tau - c$ with σ^*



Step 1: searcher gets at least $\tau - c$ with σ^*



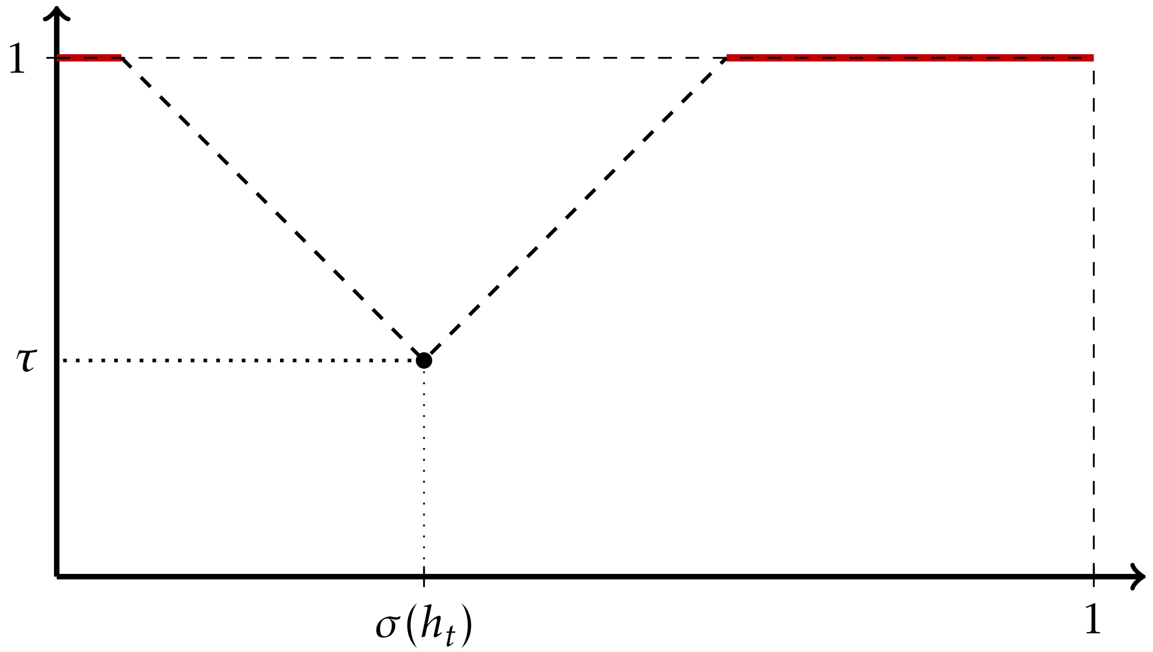
Proof steps

Step 0: guess worst-case $q \in Q_h$ for σ^* when $S_h = [a, 1]$

Step 1: verify this guess, i.e., for any $q' \in Q_h$, $p\left(h_q^{\sigma^*}\right) \leq p\left(h_{q'}^{\sigma^*}\right)$

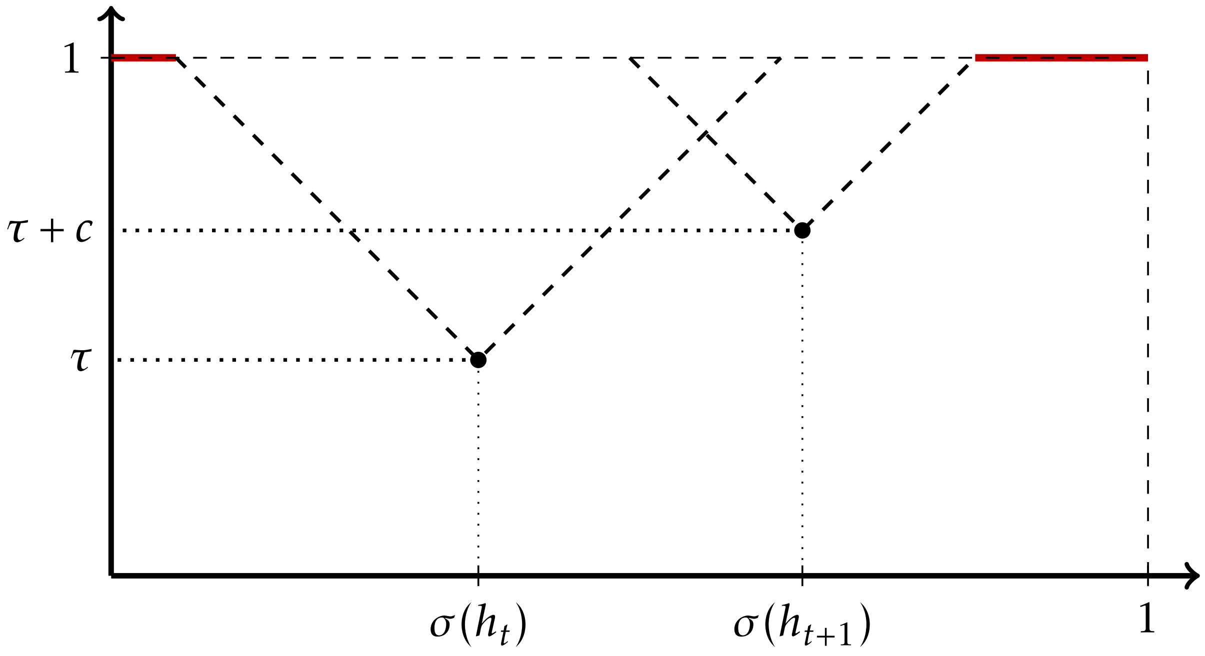
Step 2: for any $\sigma \in \Sigma$, there is a $q' \in Q_h$ with $p\left(h_{q'}^{\sigma}\right) \leq p\left(h_q^{\sigma^*}\right)$

Step 2: searcher could get $\leq \tau - c$ with σ



Payoff to stopping:
$\tau - c$

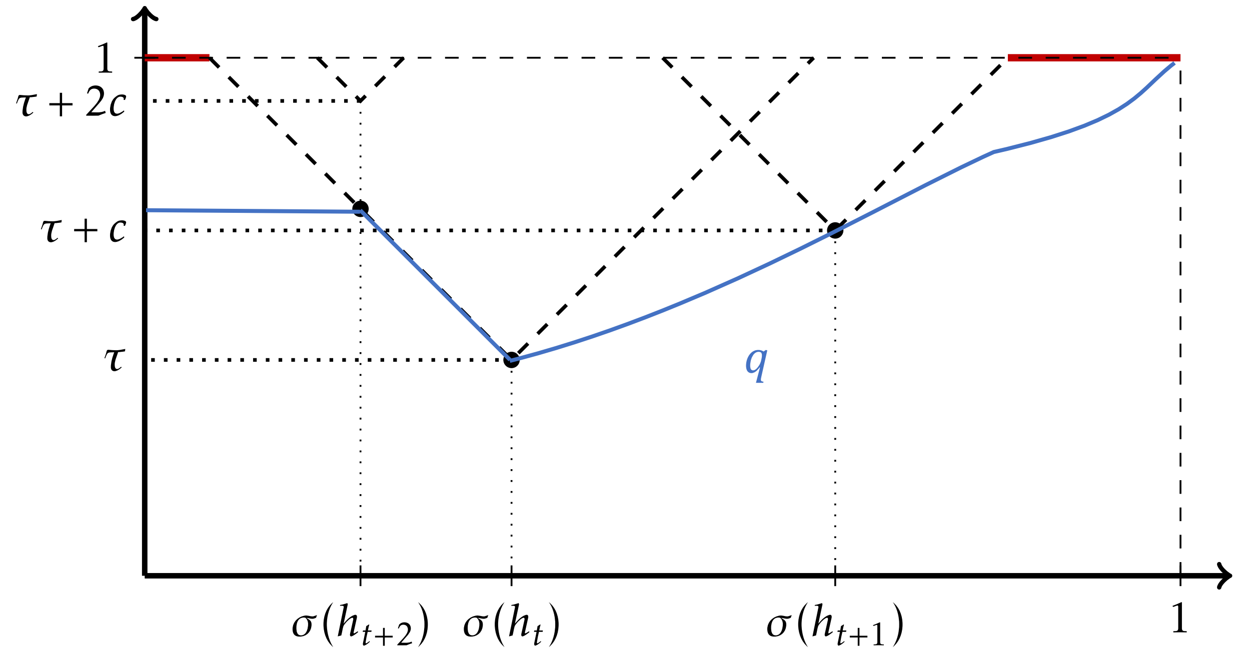
Step 2: searcher could get $\leq \tau - c$ with σ



Payoff to stopping:

$\tau + c \; - 2c = \tau - c$

Step 2: searcher could get $\leq \tau - c$ with σ



Payoff to stopping:

$< \tau + 2c - 3c = \tau - c$

With additional searches, payoff:

$\leq 1 - 4c < \tau - c$

Proof steps

Step 0: guess worst-case $q \in Q_h$ for σ^* when $S_h = [a, 1]$

Step 1: verify this guess, i.e., for any $q' \in Q_h$, $p\left(h_q^{\sigma^*}\right) \leq p\left(h_{q'}^{\sigma^*}\right)$

Step 2: for any $\sigma \in \Sigma$, there is a $q' \in Q_h$ with $p\left(h_{q'}^{\sigma}\right) \leq p\left(h_q^{\sigma^*}\right)$

Step 3: if $S_h = [a, b] \sqcup [c, 1]$, stopping is optimal