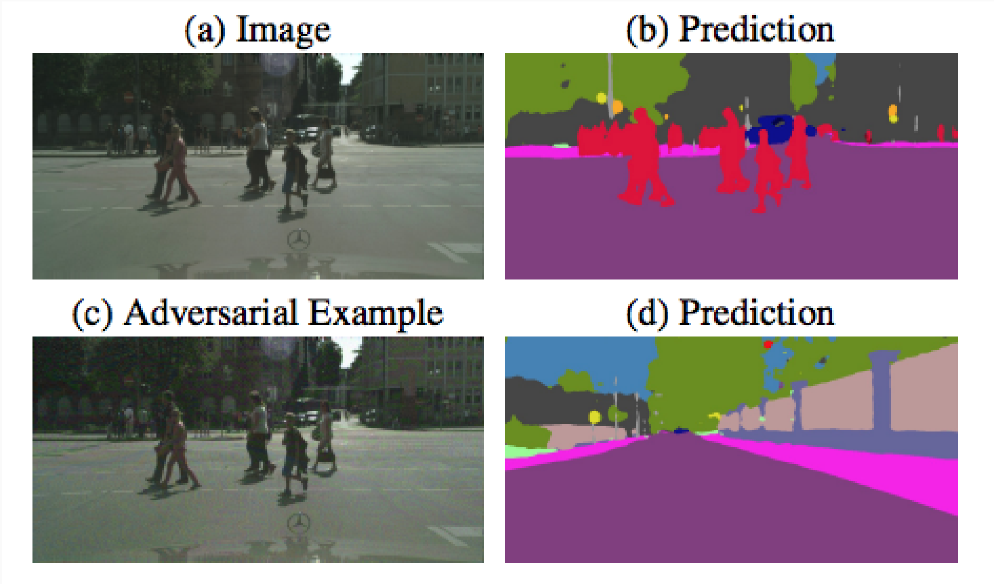
Decision Analytic Support in Non-Cooperative Sequential Games

Roi Naveiro, Tahir Ekin, Aberto Torres, David Ríos
roi.naveiro@icmat.es | Inst. of Mathematical Sciences (ICMAT-CSIC)
Madrid | ADA 2022

Sequential Defense Attack Games

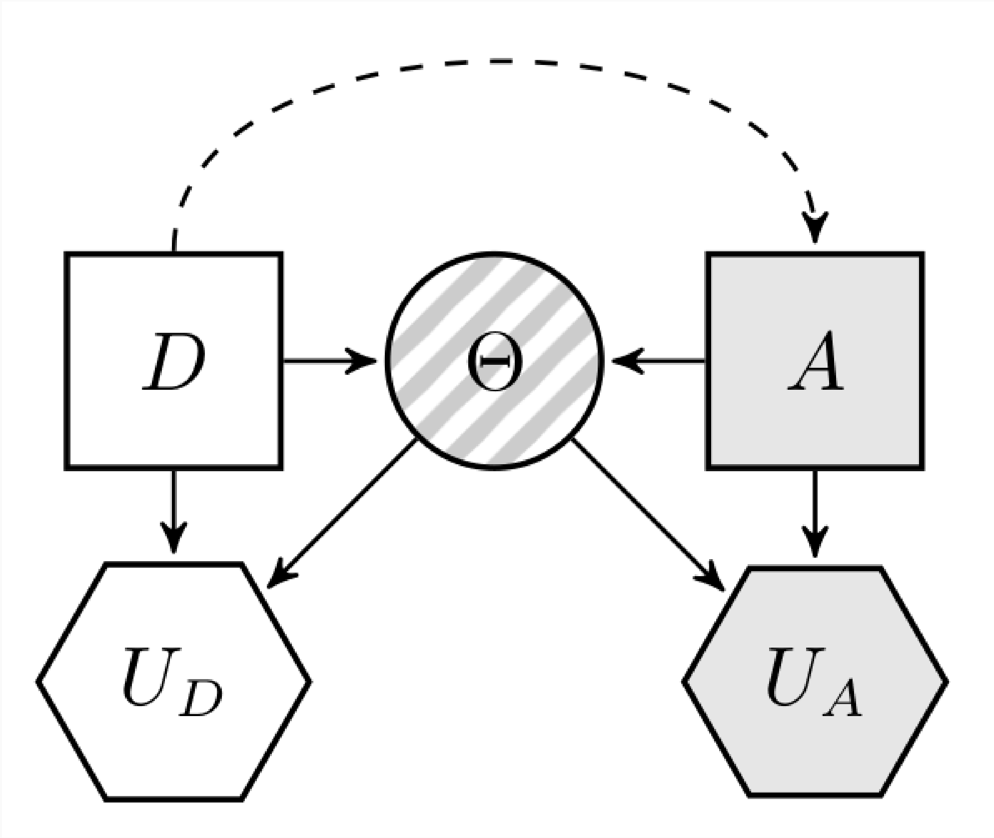


- Gaining Importance due to the raise of AML!
- Classical Decision Makers, Humans: **discrete** and **low dimensional** decision spaces.
- New Decision Makers, Algorithms: **continuous** and **high dimensional** decision spaces.

New Solution techniques

- Forget about (general) analytic solutions!
- Must work with uncertain outcomes
- Must acknowledge uncertainty about adversary
- We propose a **Simulation-based** solution approach:
    - Solves general security games, with uncertain outcomes, complete and incomplete information
    - Explain it for Sequential Defend-Attack games under incomplete information

Seq. Games with Uncertain Outcomes



Game theoretic approach

- **Common Knowledge Assumtion**: the Defender knows the Attacker's probabilities and utilities.
- Compute expected utilities.

$$\psi_A(a,d) = \int u_A(a,\theta)\, p_A(\theta|d,a)\ \mathrm{d}\theta \quad \text{and} \quad \psi_D(d,a) = \int u_D(d,\theta)\, p_D(\theta|d,a)\, \mathrm{d}\theta.$$

- Attacker's best response to defense d

$$a^*(d) = \arg\max_{a \in \mathcal{A}}\ \psi_A(d,a)$$

- Defender's optimal action

$$d^*_{\mathrm{GT}} = \arg\max_{d \in \mathcal{D}}\ \psi_D(d, a^*(d)).$$

- $\left[d^*_{\mathrm{GT}},\ a^*(d^*_{\mathrm{GT}})\right]$ is a **Nash equilibrium** and a **sub-game perfect equilibrium**.

ARA approach

- Weaken Common Knowledge Assumption: the Defender **does not know** (u_A, p_A).
- We need $p_D(a|d)$!
- Then, $d^*_{\text{ARA}} = \arg\max_{d \in \mathcal{D}} \psi_D(d)$, where

$$\psi_D(d) = \int \psi_D(a, d)\, p_D(a|d)\, \mathrm{d}a = \int \left[\int u_D(d, \theta)\, p_D(\theta|d, a)\, \mathrm{d}\theta \right] p_D(a|d)\, \mathrm{d}a,$$

ARA approach

- To elicitate $p_D(a|d)$, Defender analyses Attacker's problem.
- Model uncertainty about (u_A, p_A) through distribution $F = (U_A, P_A)$.
- Induces distribution over attacker's expected utility $\Psi_A(a, d) = \int U_A(a, \theta) P_A(\theta | a, d) \, \mathrm{d}\theta$.
- And $A^*(d) = \arg\max_{x \in \mathcal{A}} \Psi_A(x, d)$
- Then,

$$p_D(A \leq a | d) = \mathbb{P}_F \left[A^*(d) \leq a \right],$$

ARA approach

- In practice, discretize decision set
- Draw J samples $\left\{\left(P_A^i, U_A^i\right)\right\}_{i=1}^J$ from F and

$$\hat{p}_D(a|d) \approx \frac{\#\{a = \arg\max_{x \in \mathcal{A}} \ \Psi_A^i(x,d)\}}{J},$$

- ARA solution is a **Bayes-Nash Eq.** (in sequential games)

MC solution method

input: J

for $d \in \mathcal{D}$ **do**

  for $i = 1$ **to** J **do**

    Sample $u_A^i(a,\theta) \sim U_A(a,\theta)$

    Sample $p_A^i(\theta \mid a, d) \sim P_A(\theta \mid d, a)$

    Compute $a_i^*(d)$ as $\arg\max_a \int u_A^i(a,\theta) p_A^i(\theta \mid a, a)\, \mathrm{d}\theta$

  $\hat{p}_D(A^* = a \mid d) = \frac{1}{J}\sum_{i=1}^{J} I[a_i^*(d) = a]$

Solve $\max_d \int\int u_D(d,\theta) p_D(\theta \mid a, d) \hat{p}_D(A^* = a \mid d)\, \mathrm{d}\theta\, \mathrm{d}a$

- Requires generating $|\mathcal{D}| \times (|\mathcal{A}| \times Q \times J + P)$ samples.

APS - Idea 1

- Assume we can sample from $p_D(d|a)$
- Max expected utility

$$d^*_{\text{ARA}} = \arg\max_d \int\int u_D(d,\theta) \cdot p_D(\theta|d,a) \cdot p_D(d|a) d\theta da$$

- Define

$$\pi_D(d,a,\theta) \propto u_D(d,\theta) \cdot p_D(\theta|d,a) \cdot p_D(d|a)$$

- Mode of marginal $\pi_D(d)$ is d^*_{ARA} !

APS - Idea 2

- Flat expected utilities, complicates mode identification
- Define

$$\pi_D^H(d, \theta_1, \ldots, \theta_H, a_1, \ldots, a_H) \propto \prod_{i=1}^{H} u_D(d, \theta_i) \cdot p_D(\theta_i | d, a_i) \cdot p_D(a_i | d)$$

- Marginal more peaked around max!

$$\pi_D^H(d) \propto \left[\int \int u_D(d, \theta) \cdot p_D(\theta | d, a) \cdot p_D(d | a) d\theta da \right]^H$$

APS - Implementation

- Sample from $\pi(d, \theta_1, \theta_2, \ldots, \theta_H, a_1, \ldots, a_H)$ using MCMC.
- Find mode of d samples.

1. State of the Markov chain is $(d, \theta_1, \ldots, \theta_H, a_1, \ldots, a_H)$;
2. $\tilde{d} \sim g(\cdot|d)$;
3. $\tilde{a}_i \sim p_D(a|\tilde{d})$ for $i = 1, \ldots, H$;
4. $\tilde{\theta}_i \sim p_D(\theta|\tilde{d}, \tilde{a}_i)$ for $i = 1, \ldots, H$;
5. Accept $\tilde{x}, \tilde{\theta}_1, \ldots, \tilde{\theta}_H, \tilde{a}_1, \ldots \tilde{a}_H$ with probability

$$\min \left\{ 1, \frac{g(d|\tilde{d})}{g(\tilde{d}|d)} \cdot \prod_{i=1}^{H} \frac{u_D(\tilde{d}, \tilde{\theta}_i)}{u_D(d, \theta_i)} \right\}$$

6. Repeat

- Embed this MCMC within an annealing schedule that increases H!

APS for ARA - $p_D(a|d)$

- For given d, random augmented distribution $\Pi_A(a,\theta|d) \propto U_A(a,\theta)P_A(\theta|d,a)$,
- Marginal $\Pi_A(a|d) = \int \Pi_A(a,\theta|d)d\theta$, proportional to A's random expected utility $\Psi_A(d,a)$.
- Random optimal attack $A^*(d)$ coincides a.s. with mode of $\Pi_A(a|d)$.
- Then:

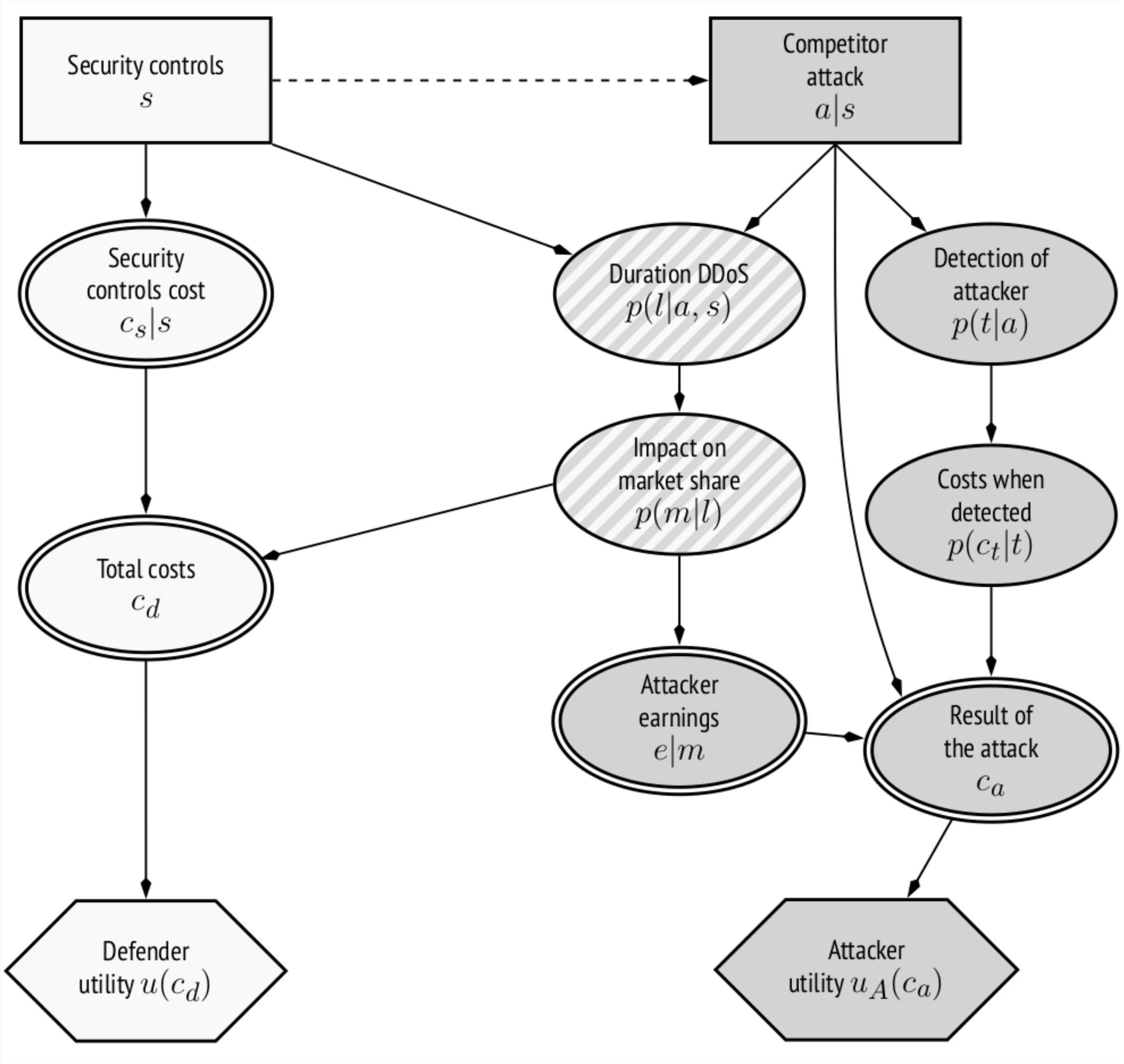
1. $u_A(a,\theta) \sim U_A(a,\theta)$ and $p_A(\theta|d,a) \sim P_A(\theta|d,a)$
2. Build $\pi_A(a,\theta|d) \propto u_A(a,\theta)p_A(\theta|d,a)$ which is a sample from $\Pi_A(a,\theta|d)$.
3. Find $\mathbf{mode}[\pi_A(a|d)]$ which is a sample of $A^*(d)$, whose distribution is $\mathbb{P}_F\left[A^*(d) \leq a\right] = p_D(a \leq d)$.

APS vs MC

- MC requires $|\mathcal{D}| \times (|\mathcal{A}| \times Q \times J + P)$ samples
- APS requires at most $N \times (2M + 5) + 2M + 4$ samples
- Simple game with continuos decision sets
- Several in parallel
- Compute min number of samples s.t. 90% solutions coincide with truth (to required precision)

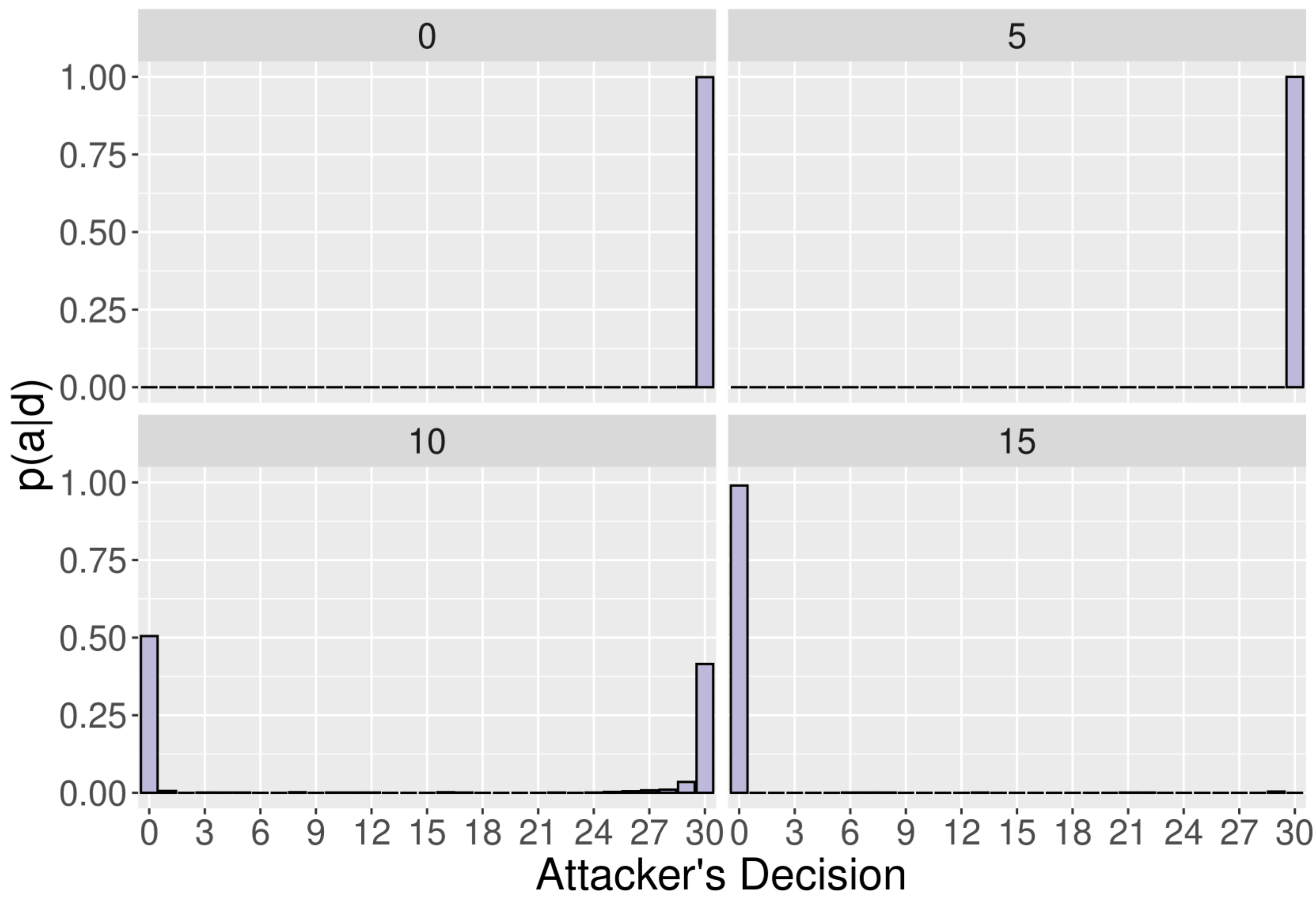
		Samples		Power		
Precision	Algorithm	Outer	Inner	Outer	Inner	Time (s)
0.1	MC	1000	100	-	-	0.007
	APS	60	100	900	20	0.240
0.01	MC	717000	100	-	-	13.479
	APS	300	100	6000	100	2.461

Application



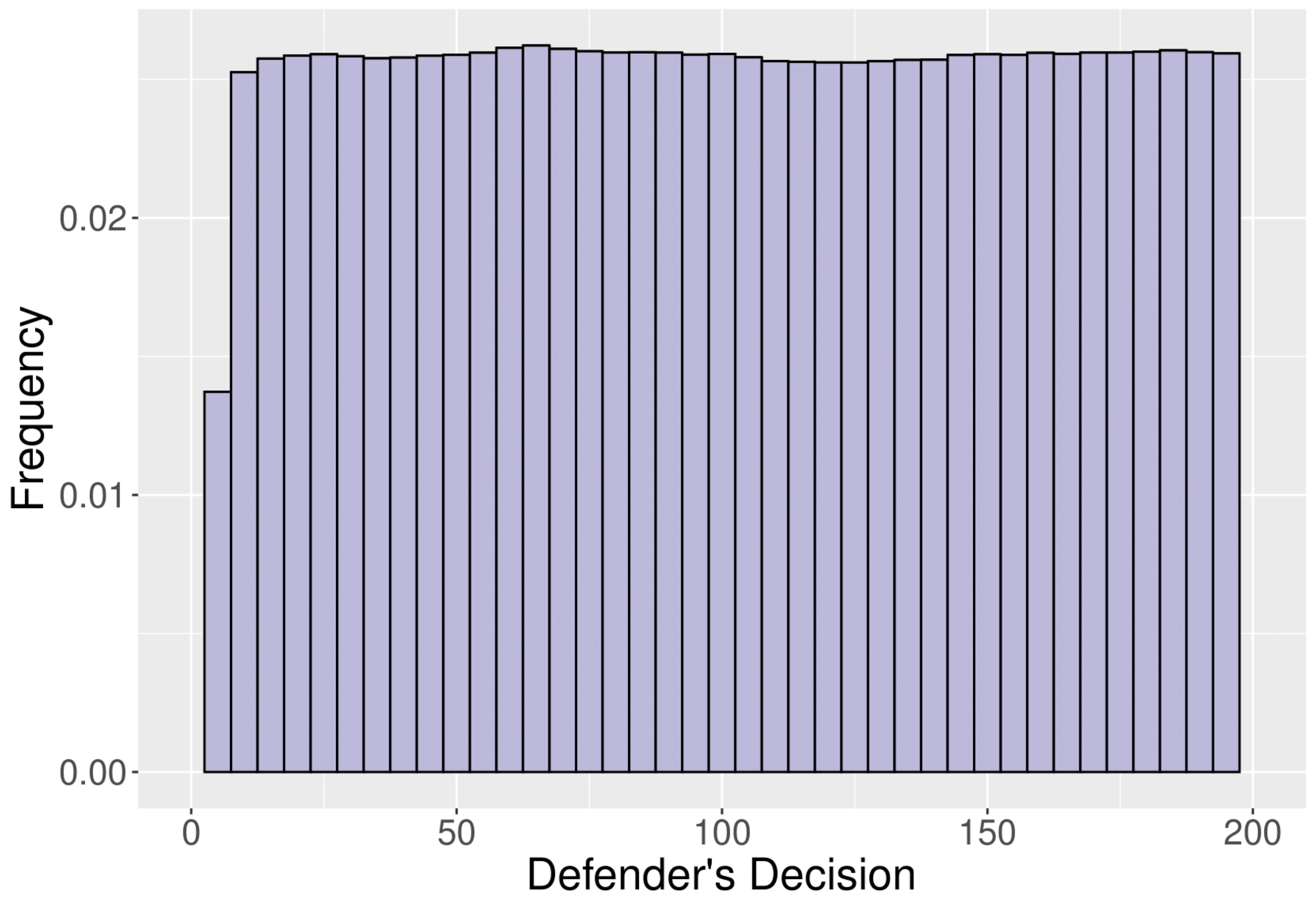
Application

- Elicited probability $p(a|d)$ for some security controls.

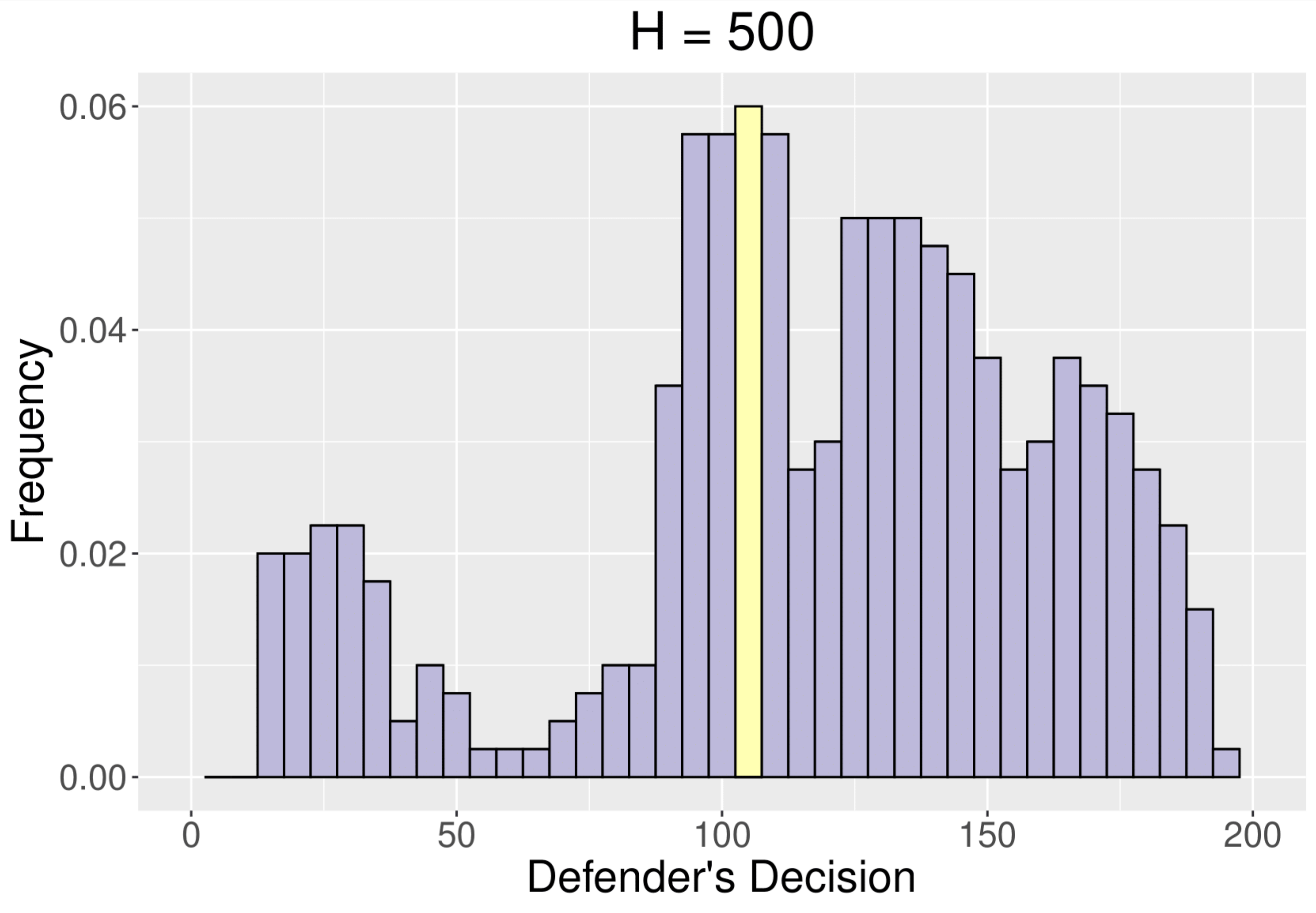


Application

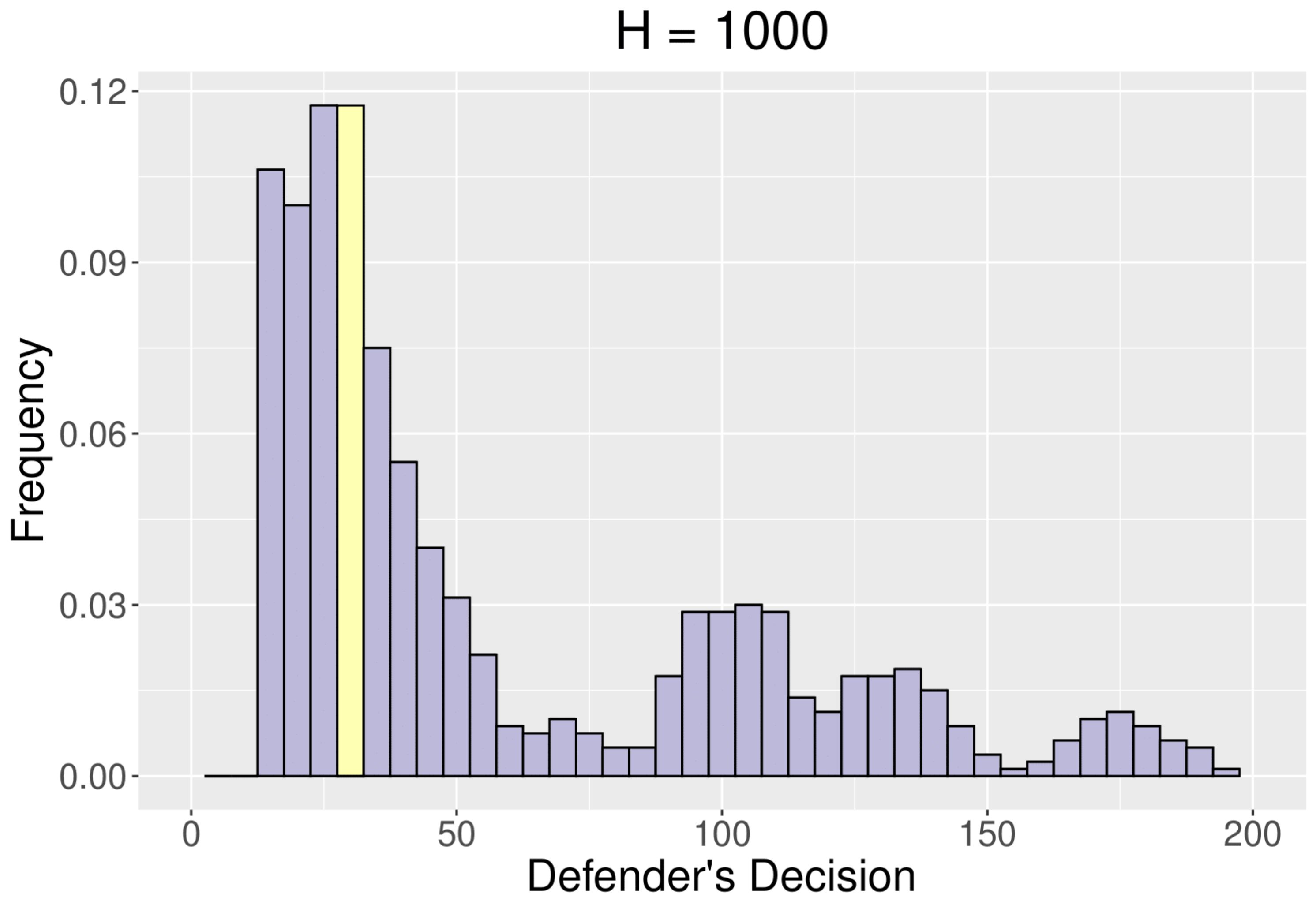
- Histogram of samples of security controls.



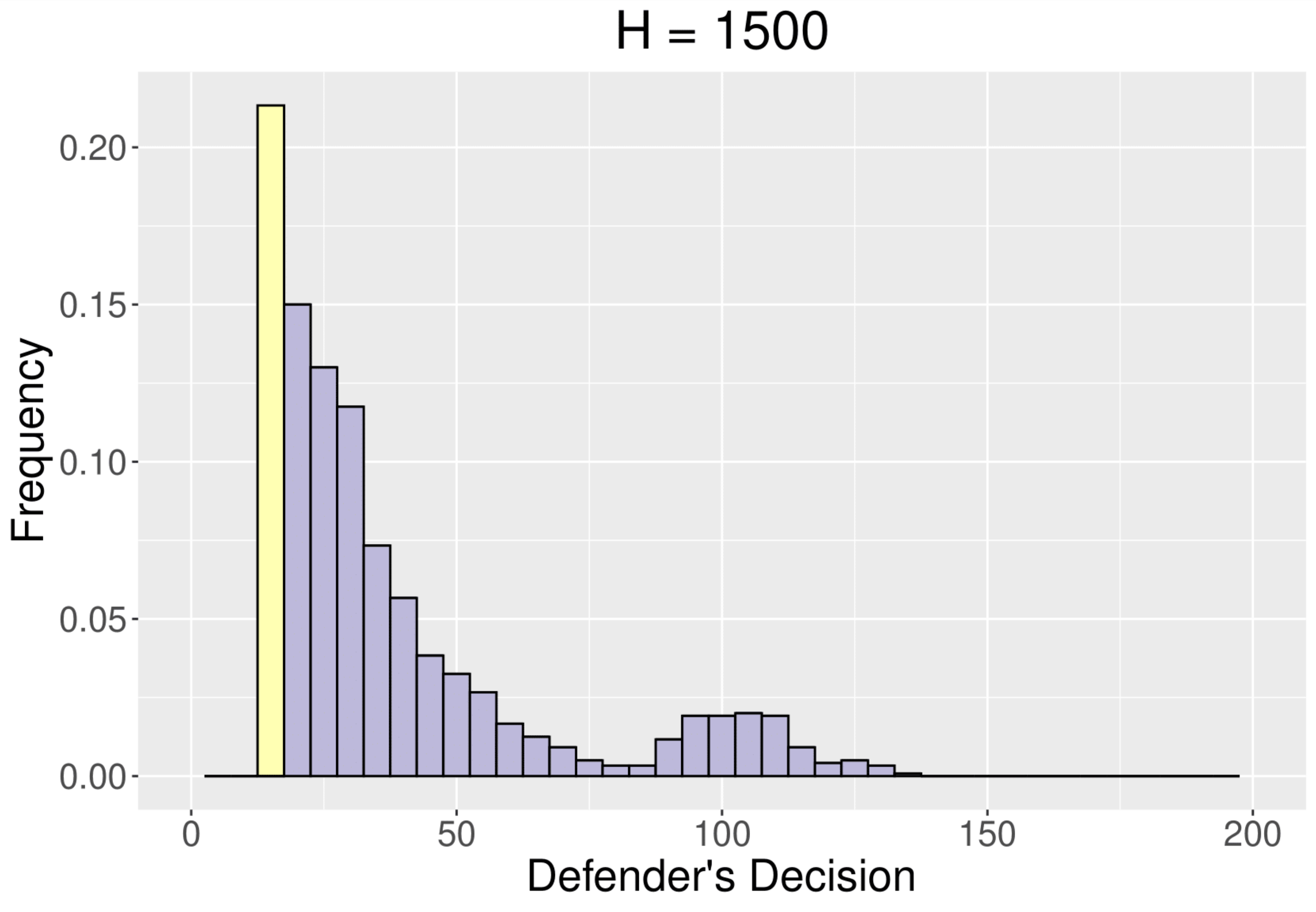
Application - Increasing H



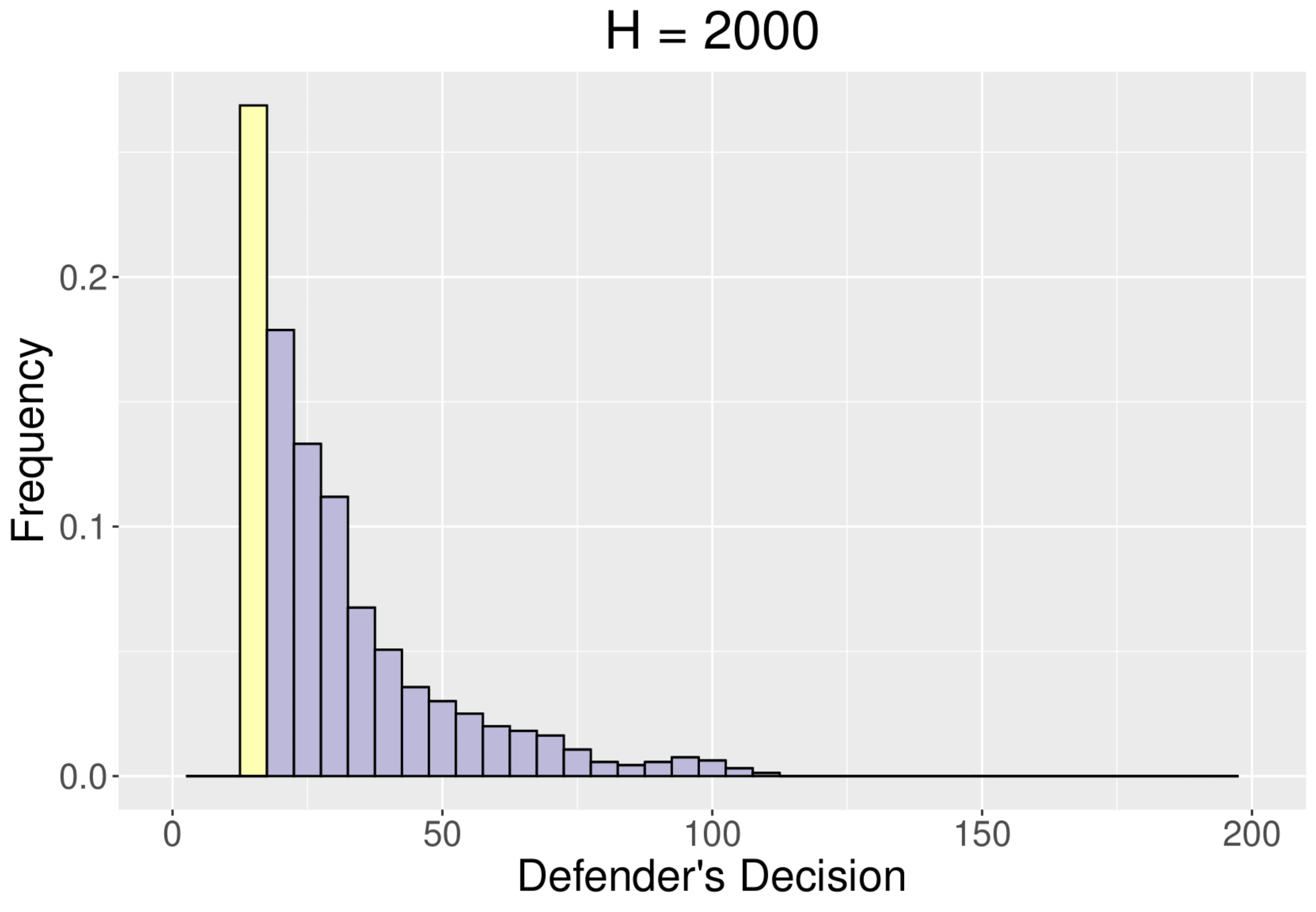
Application - Increasing H



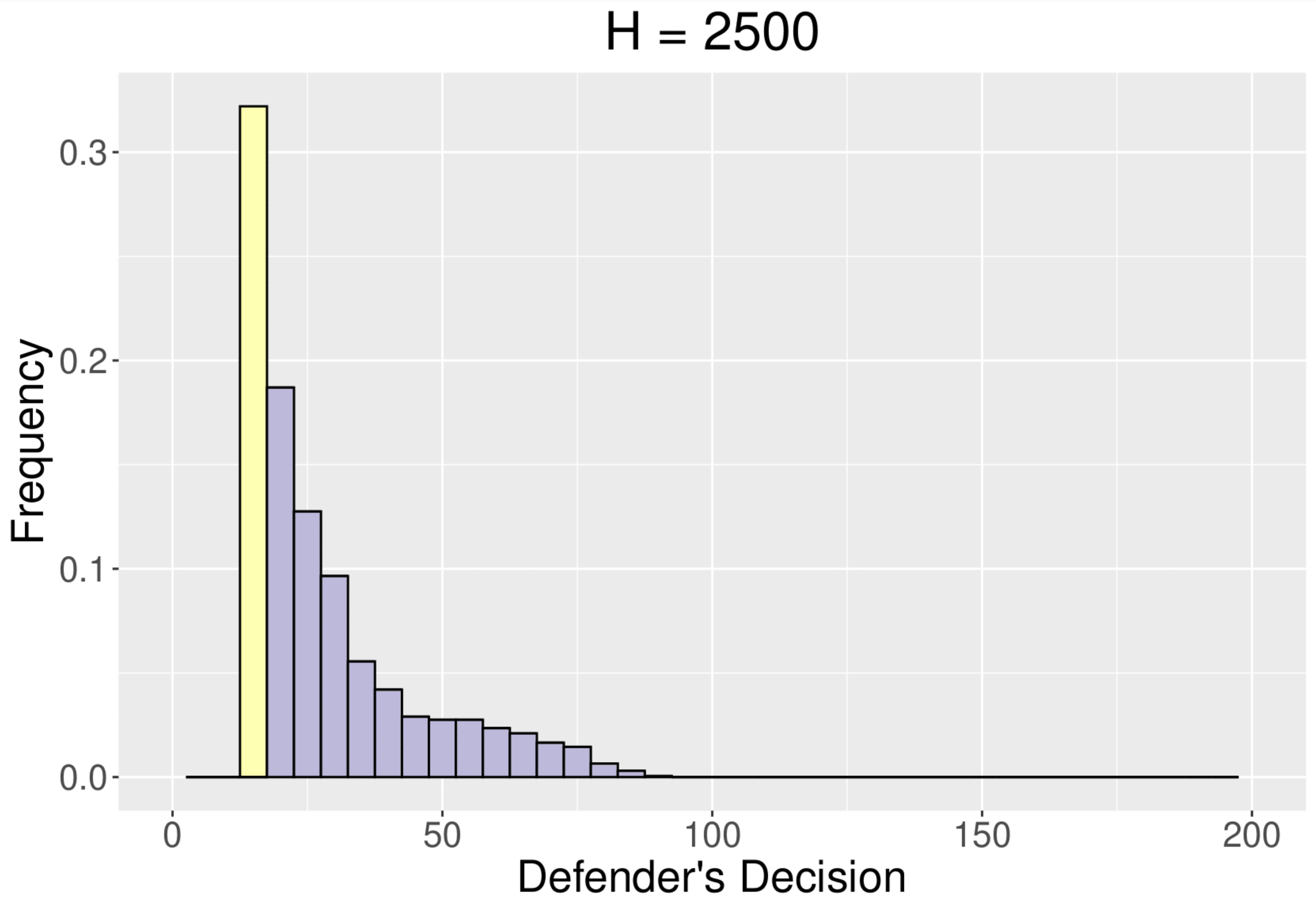
Application - Increasing H



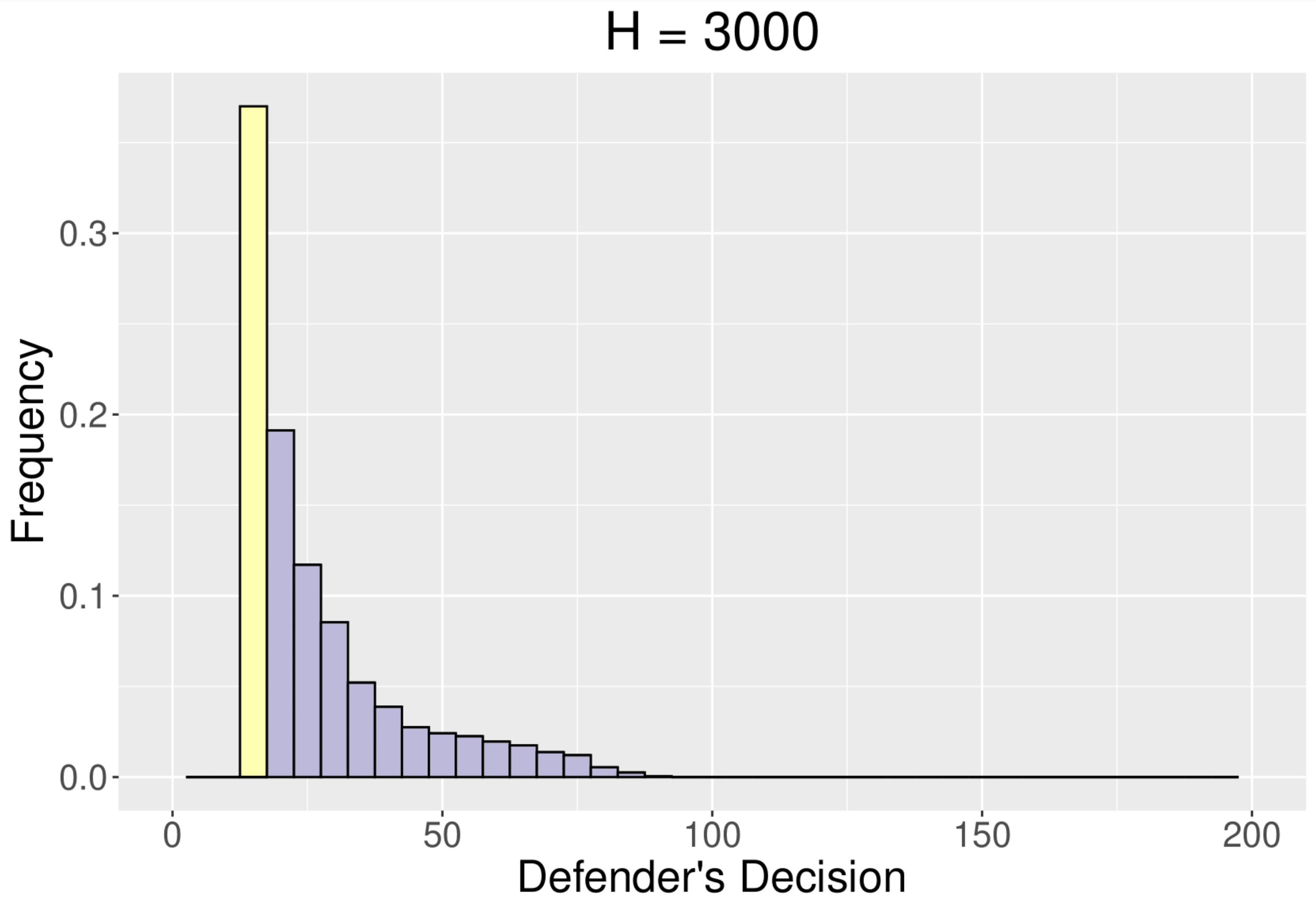
Application - Increasing H



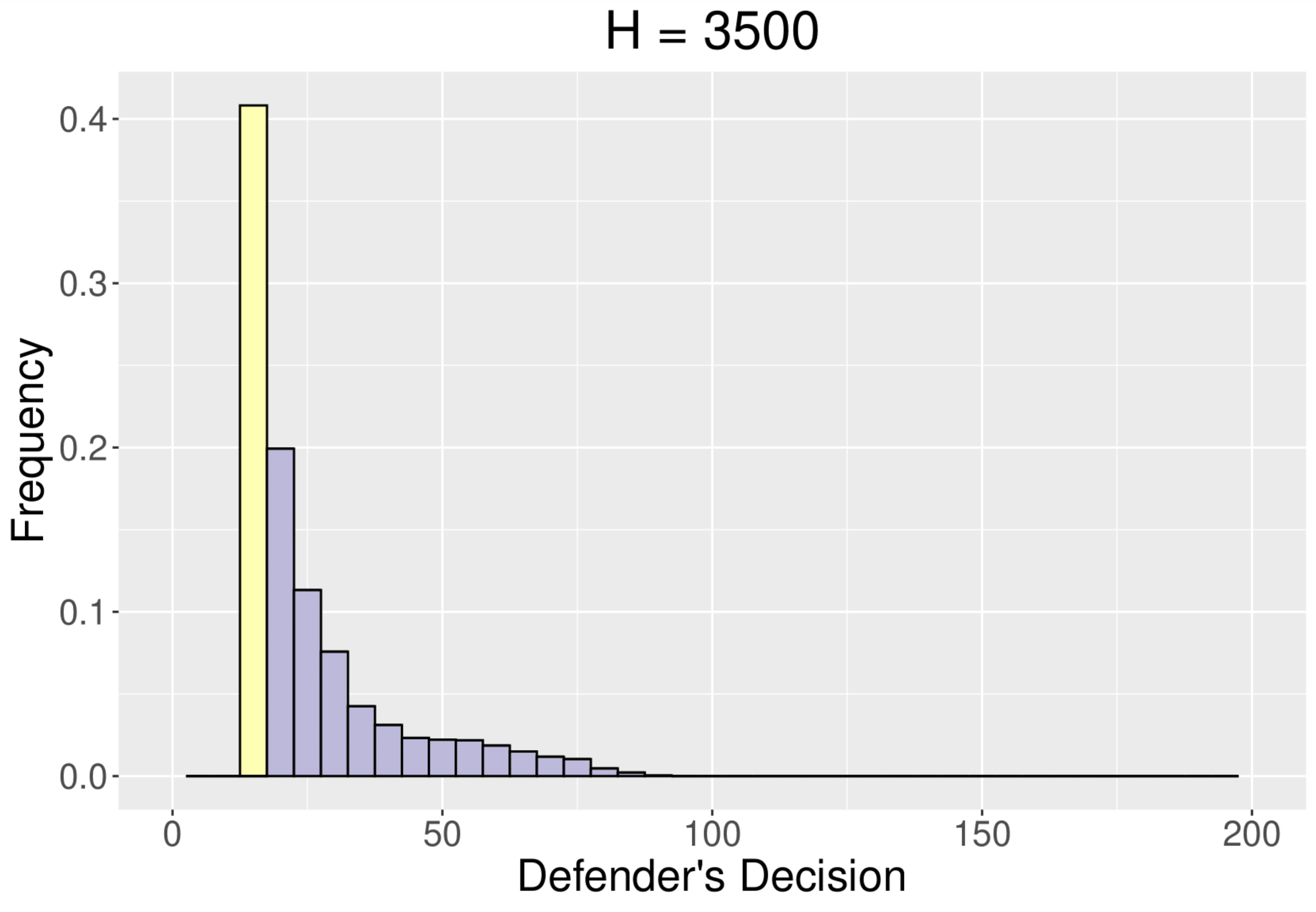
Application - Increasing H



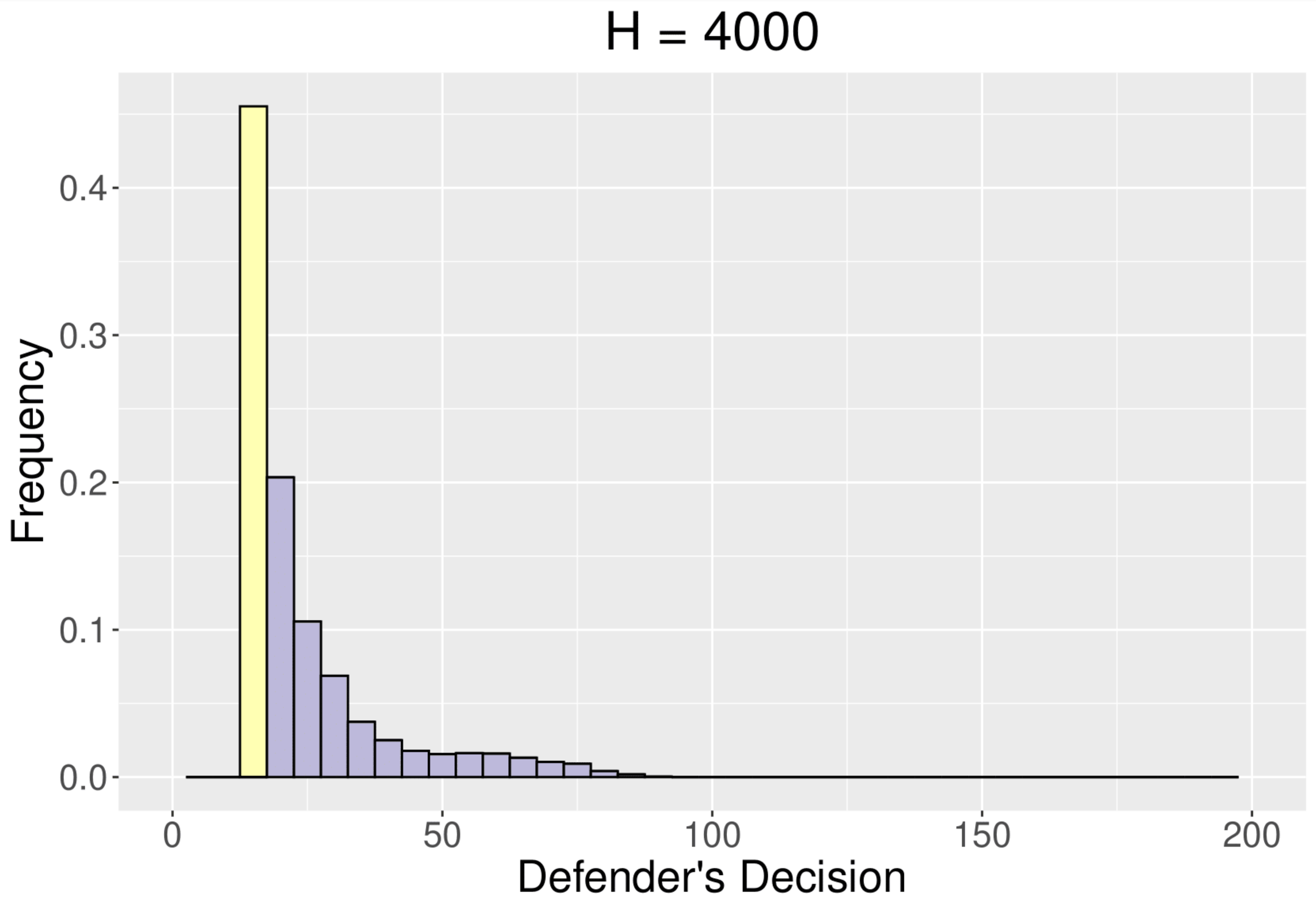
Application - Increasing H



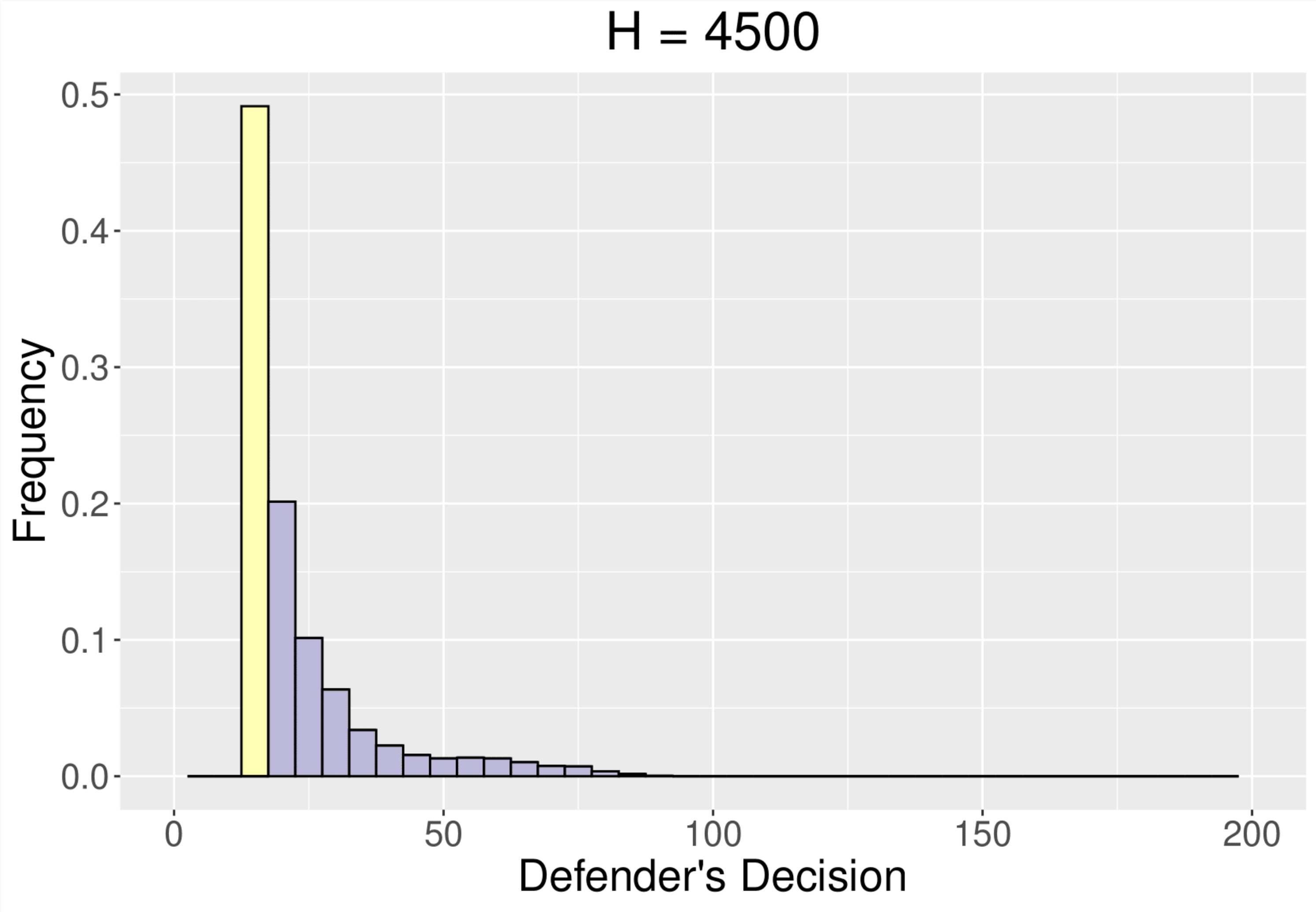
Application - Increasing H



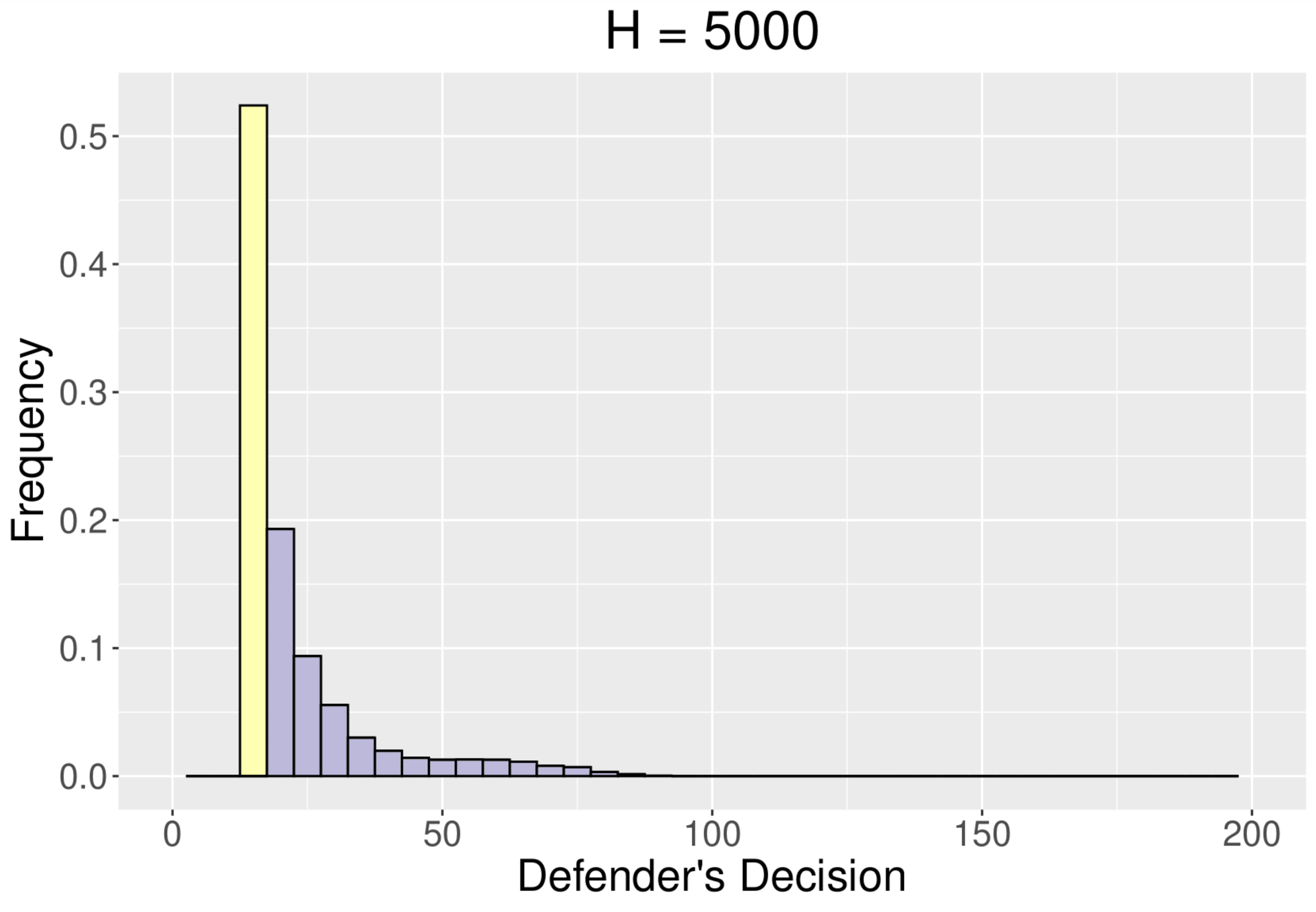
Application - Increasing H



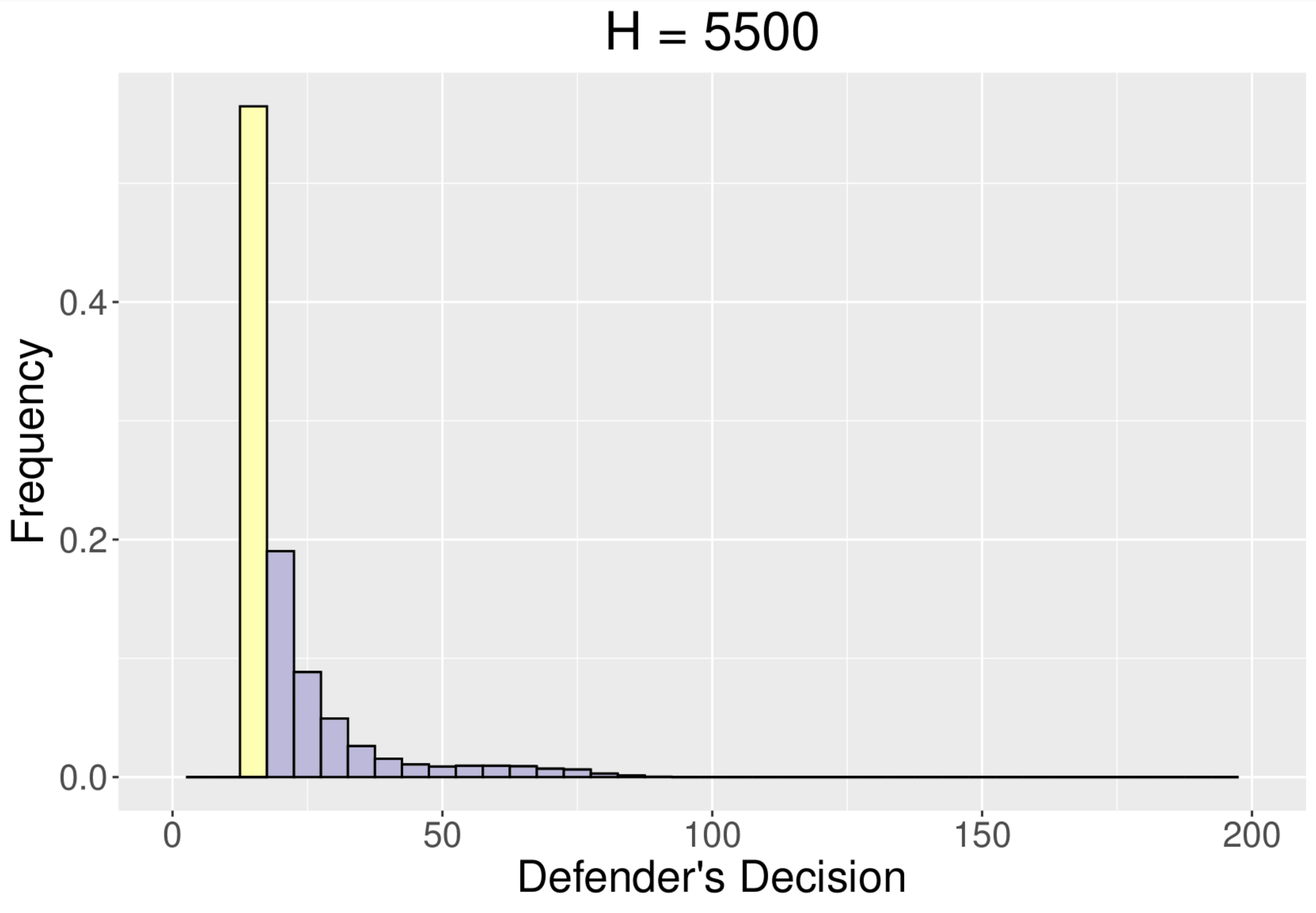
Application - Increasing H



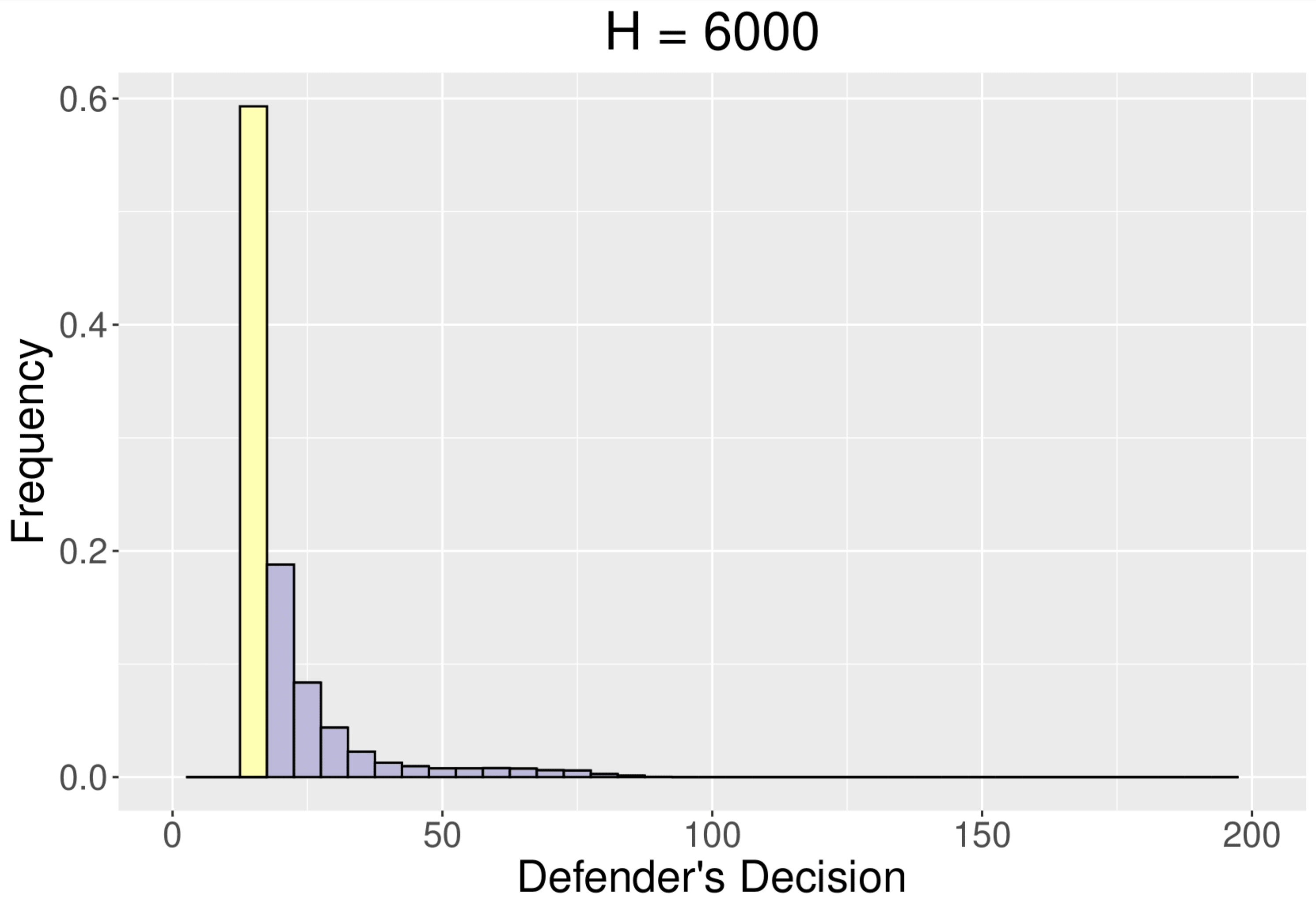
Application - Increasing H



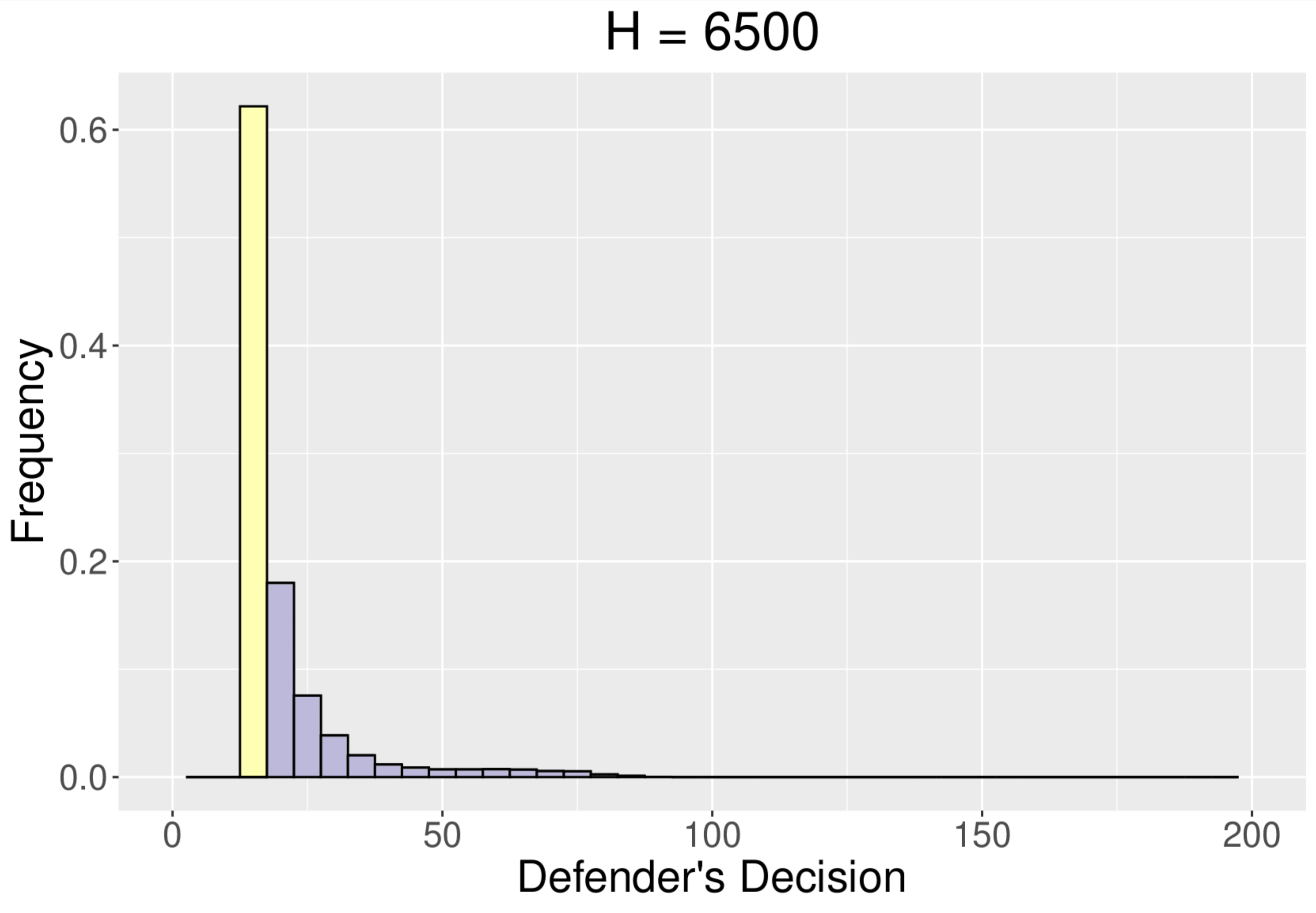
Application - Increasing H



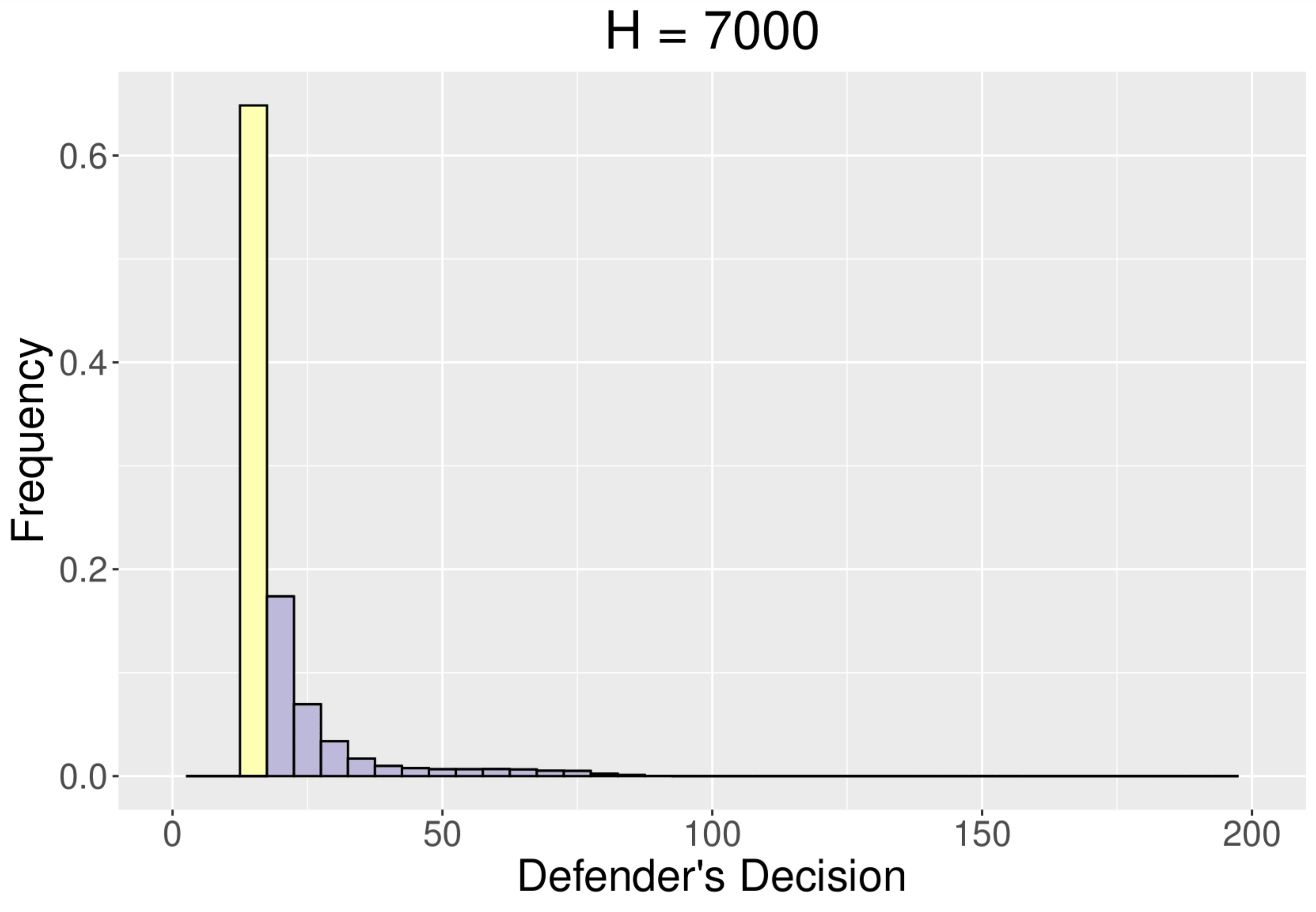
Application - Increasing H



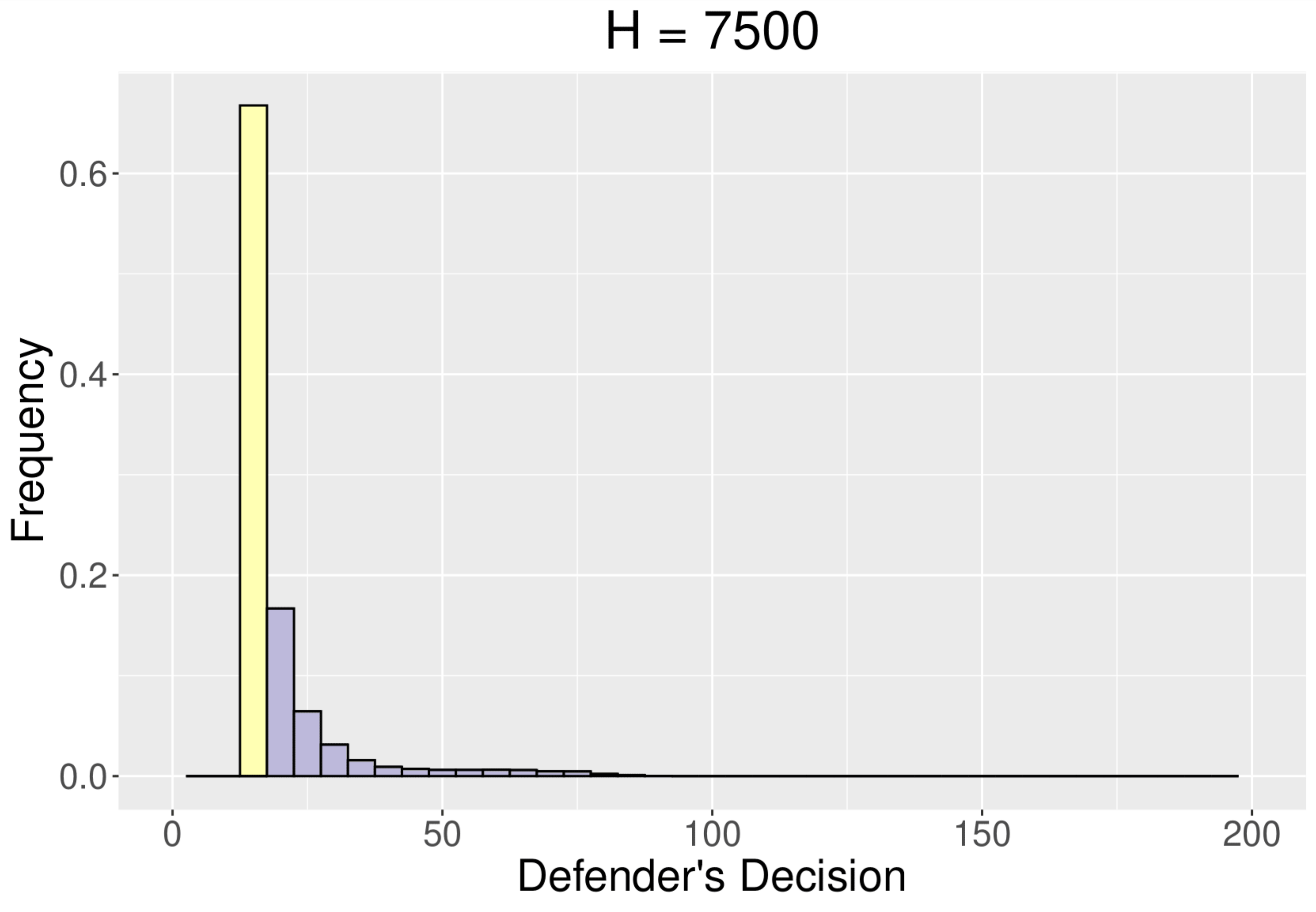
Application - Increasing H



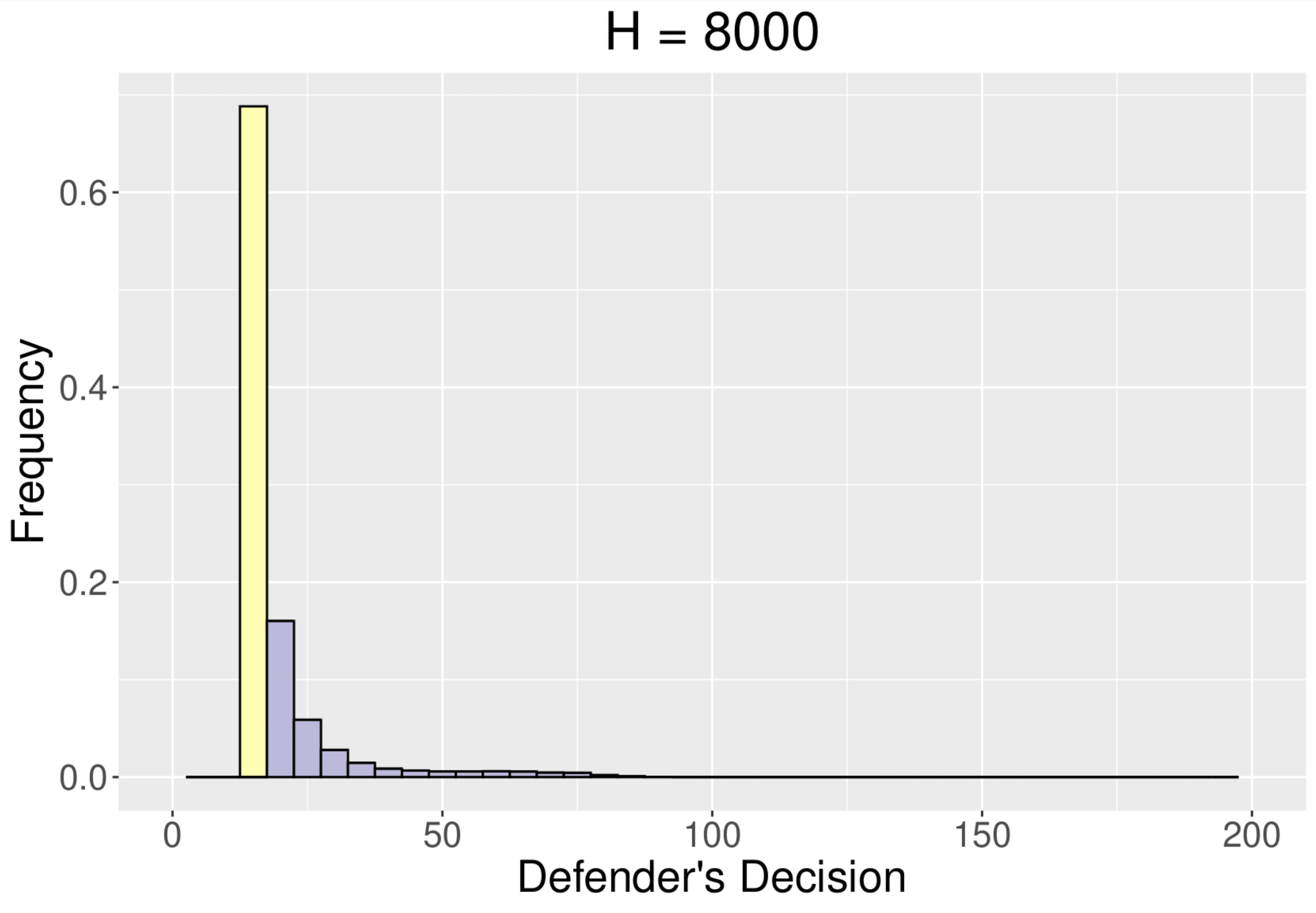
Application - Increasing H



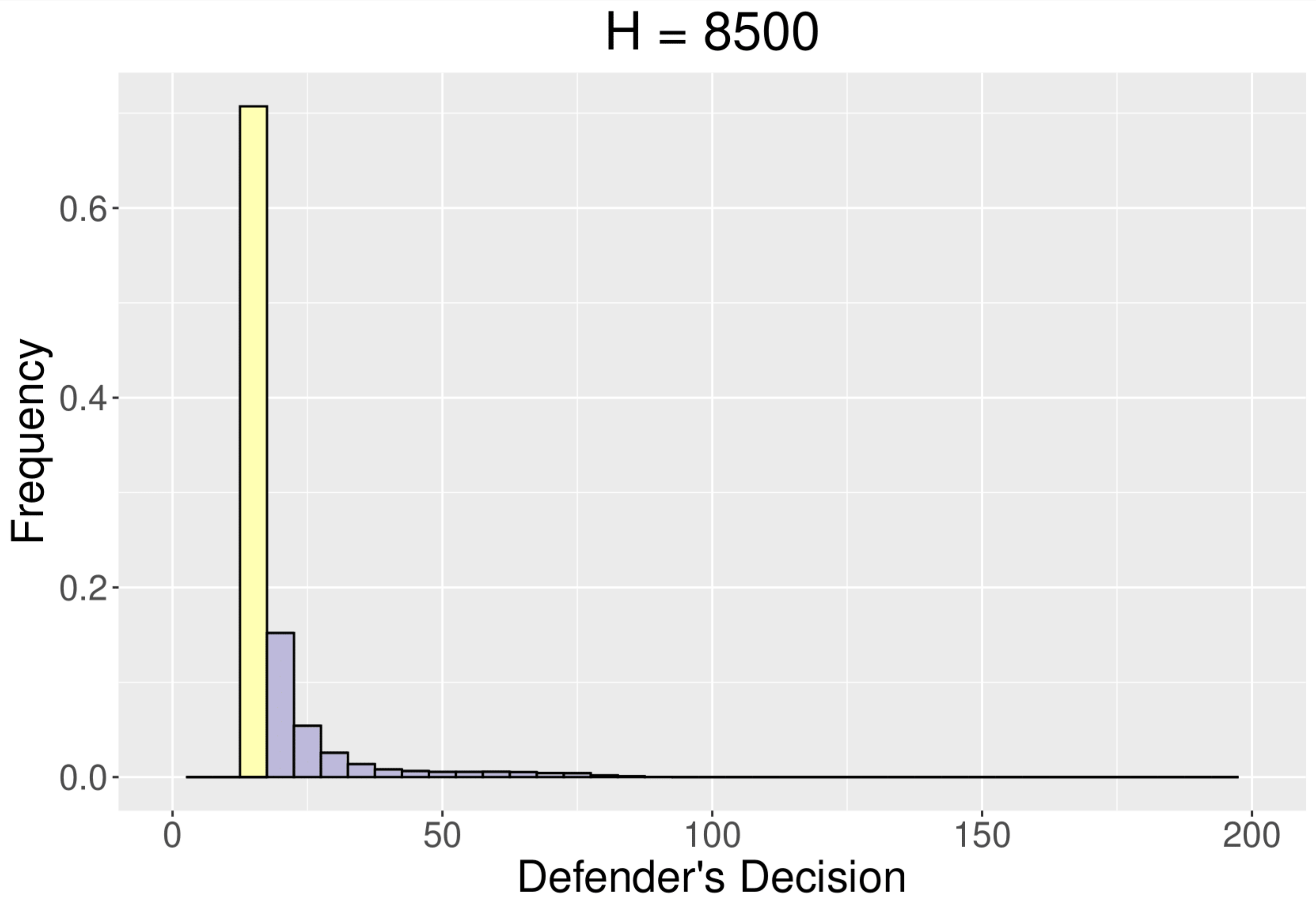
Application - Increasing H



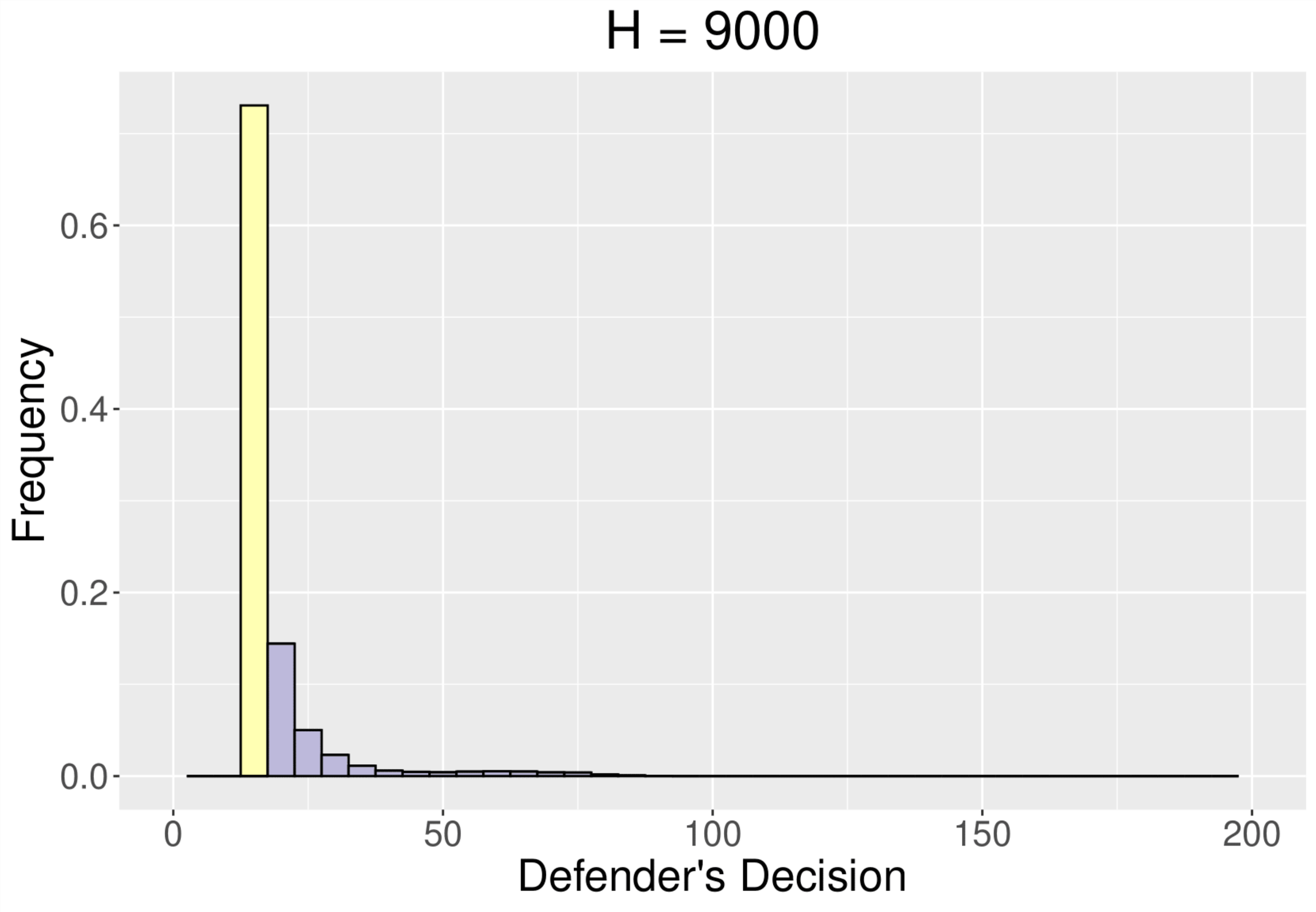
Application - Increasing H



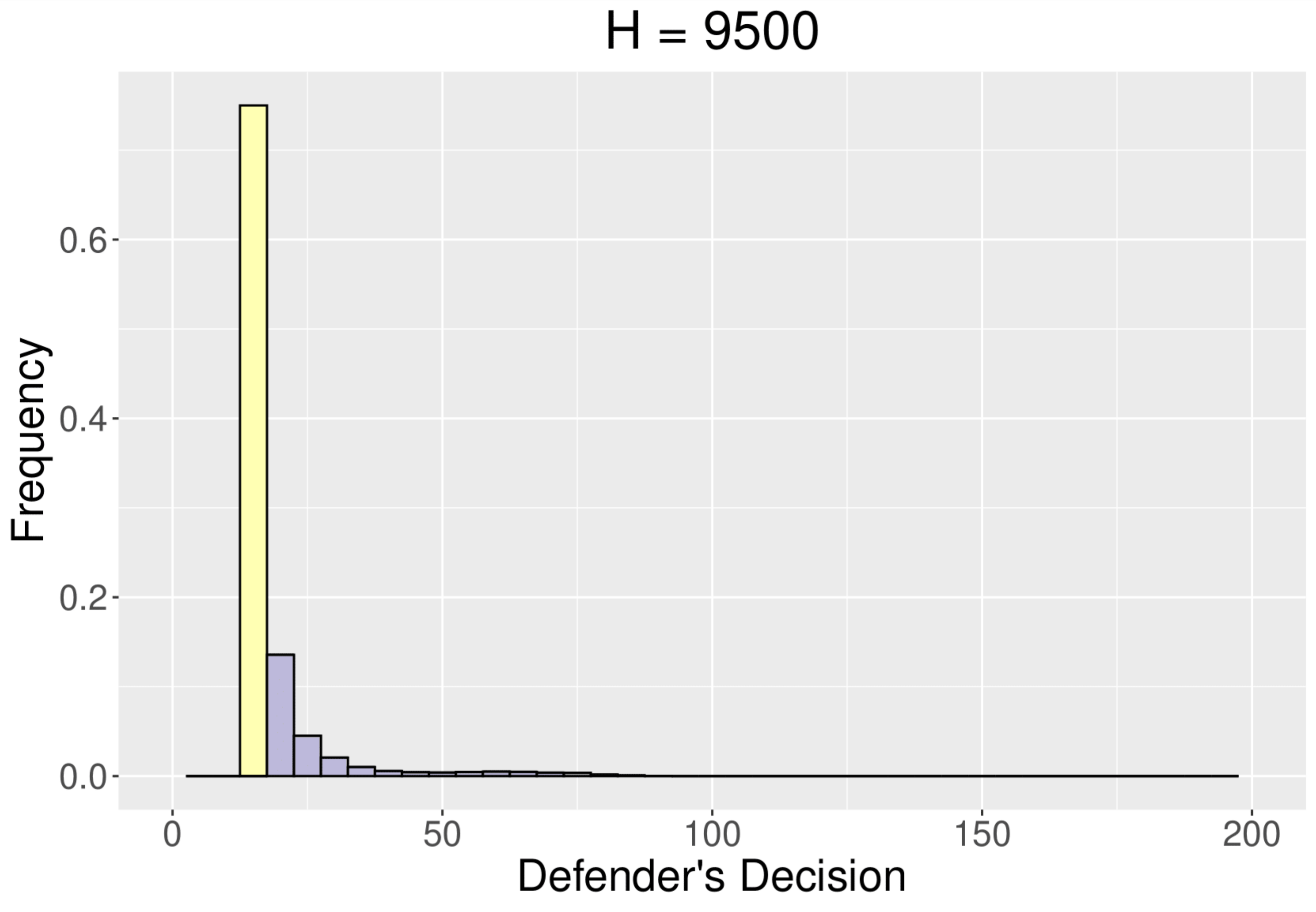
Application - Increasing H



Application - Increasing H



Application - Increasing H



Conclusions

- APS for games, both standard and ARA.
- APS better when cardinality of decision spaces is big (or spaces are continuous).
- Suggested **algorithmic approach**
  1. Use MC for broad exploration of decision space.
  2. Use APS within regions of interest to get refined solutions.

Thank you!!

Website roinaveiro.github.io/

Email roi.naveiro@icmat.es

GitHub github.com/roinaveiro