Value of Information in Optimal Flow-Level Scheduling of Users with Markovian Time-Varying Channels

Peter Jacko*

Performance 2011, October 18

*BCAM — Basque Center for Applied Mathematics, Spain

Motivation: Wireless Downlink

- Channel conditions vary due to fading
- Geometric-sized jobs
- Channel conditions independent across users
- Markovian evolution of channel conditions

- Base station can serve M users per slot

Talk Outline

- Flow-level MDP model
- Relaxation of the resource allocation constraint
- Optimal index policy and indexability
- Generalized Potential Improvement rule
  - new opportunistic scheduler
- Suboptimality evaluation by numerical experiments

Job Scheduling Problem

- Discrete time ($t = 0, 1, 2, \ldots$), preemptive service
- Jobs $k = 1, 2, \ldots$ with size B_k (in bits) arrive randomly
  - ▷ c_k = cost of waiting for job k
  - ▷ Gilbert-Elliot channel quality conditions $\mathcal{N}'_k := \{\mathsf{B}, \mathsf{G}\}$

$$\boldsymbol{Q}_k = \begin{array}{c} \\ \mathsf{B} \\ \mathsf{G} \end{array} \begin{array}{c} \begin{array}{cc} \mathsf{B} & \mathsf{G} \end{array} \\ \begin{pmatrix} q_{k,\mathsf{B},\mathsf{B}} & q_{k,\mathsf{B},\mathsf{G}} \\ q_{k,\mathsf{G},\mathsf{B}} & q_{k,\mathsf{G},\mathsf{G}} \end{pmatrix} \end{array}$$

  - ▷ service rate $0 \leq s_{k,B} \leq s_{k,G}$ bits per second
- Minimize total waiting cost while serving M jobs/slot

Markov Decision Processes Model

- Job/user/channel k is defined by
  - ▷ action space $\mathcal{A} := \{0, 1\}$
  - ▷ departure probability
    $\mu_{k,n} = \min\{1, 1 - (1 - 1/\mathbb{E}[B_k])^{\varepsilon s_{k,n}}\}$
  - ▷ state space $\mathcal{N}_k := \{0, B, G\}$
  - ▷ expected one-period capacity consumption $\boldsymbol{W}_k^a := a$
  - ▷ expected one-period reward $\boldsymbol{R}_k^a$
  - ▷ one-period transition probability matrix $\boldsymbol{P}_k^a$
- State process $X_k(t) \in \mathcal{N}_k$
- Action process $a_k(t) \in \mathcal{A}$ – to be decided

Markov Decision Processes Model

- Expected one-period reward

$$R^1_{k,0} := 0, \qquad R^1_{k,n} := -c_k(1-\mu_{k,n}),$$
$$R^0_{k,0} := 0, \qquad R^0_{k,n} := -c_k;$$

- One-period transition probability matrices

$$\boldsymbol{P}^1_k := \begin{array}{c} \\ 0 \\ B \\ G \end{array} \begin{array}{c} \begin{array}{ccc} 0 & B & G \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ \mu_{k,B} & \widetilde{\mu}_{k,B}q_{k,B,B} & \widetilde{\mu}_{k,B}q_{k,B,G} \\ \mu_{k,G} & \widetilde{\mu}_{k,G}q_{k,G,B} & \widetilde{\mu}_{k,G}q_{k,G,G} \end{pmatrix} \end{array}$$

Resource Allocation Problem

- Formulation under the β-discounted criterion:

$$\max_{\pi \in \Pi} \sum_{k \in \mathcal{K}} \mathbb{E}^{\pi} \left[\sum_{t=0}^{\infty} \beta^t R_{k,X_k(t)}^{a_k(t)} \right]$$

subject to $$\sum_{k \in \mathcal{K}} W_{k,X_k(t)}^{a_k(t)} = M, \qquad \text{for all } t = 0, 1, 2, \ldots$$

- Analogously under the time-average criterion

- This problem is PSPACE-hard
  - ▷ intractable to solve exactly by Dynamic Programming
  - ▷ instead, we relax and decompose the problem

Resource Allocation Problem (RAP)

- Stochastic and dynamic
- There is a number of independent users
- Constraint: resource capacity at every moment
- Objective: maximize expected "reward"
- Captures the exploitation vs. exploration trade-off
  - ▷ always exploiting (being myopic) is not optimal
  - ▷ always exploring (being utopic) is not optimal
- This is a model of learning by doing!

Whittle's Relaxation

- Serve M jobs in expectation
  - ▷ infinite number of constraints is replaced by one
  - ▷ sort of perfect market assumption

$$\max_{\pi \in \Pi} \sum_{k \in \mathcal{K}} \mathbb{E}^{\pi} \left[\sum_{t=0}^{\infty} \beta^t R_{k, X_k(t)}^{a_k(t)} \right]$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \mathbb{E}^{\pi} \left[\sum_{t=0}^{\infty} \beta^t W_{k, X_k(t)}^{a_k(t)} \right] = \sum_{t=0}^{\infty} \beta^t M$$

- Provides an upper bound for RAP

Lagrangian Relaxation

- Pay cost ν for using the server
  - ▷ the constraint is moved into the objective

$$\max_{\pi \in \Pi} \sum_{k \in \mathcal{K}} \mathbb{E}^{\pi} \left[\sum_{t=0}^{\infty} \beta^t R_{k,X_k(t)}^{a_k(t)} \right] - \nu \sum_{k \in \mathcal{K}} \mathbb{E}^{\pi} \left[\sum_{t=0}^{\infty} \beta^t W_{k,X_k(t)}^{a_k(t)} \right]$$

- Also provides an upper bound for RAP
- Decomposes due to user independence into single-user parametric subproblems

Optimal Solution to Subproblems

- Theorem 1: Threshold policy is optimal
  - ▷ serve iff user-k state is above a threshold
- Theorem 2: Problem is indexable, which implies
  - ▷ if $\nu \leq \nu^*_{k,n}$, then it is optimal to serve in state n
  - ▷ if $\nu \geq \nu^*_{k,n}$, then it it optimal to wait in state n
- $\nu^*_{k,n}$ is the dynamic price (Whittle index value)
  - ▷ obtained by identifying the efficiency frontier

Index Values

- The index values for user k are

$$\nu^*_{k,G} = \frac{c_k \mu_{k,G}}{(1-\beta)},\ \nu^*_{k,B} = \frac{c_k \mu_{k,B}}{(1-\beta) + \beta q^*_{k,B,G}(\mu_{k,G} - \mu_{k,B})}$$

- Weighted harmonic mean

$$q^*_{k,B,G} := \frac{1}{\frac{1-\beta(1-\mu_{k,G})}{q_{k,B,G}} + \frac{\beta(1-\mu_{k,G})}{q^{\text{SS}}_{k,G}}}$$

- Steady-state probability for condition G

$$q^{\text{SS}}_{k,G} = \frac{q_{k,B,G}}{1 + q_{k,B,G} - q_{k,G,G}}$$

Generalized Potential Improvement Rule

- Opportunistic scheduler under time-average criterion:
  - ▷ serve M jobs with highest actual PI* index

$$\nu^*_{k,G} = \infty, \quad \nu^*_{k,B} = \frac{c_k \mu_{k,B}}{q^*_{k,B,G}(\mu_{k,G} - \mu_{k,B})}, \quad \nu^*_{k,0} = 0$$

  - ▷ tie-breaking if in the good state: $c_k \mu_{k,G}$

- Optimality in special cases
  - ▷ $M = 1$, $q_{k,\mathsf{B},\mathsf{G}} = q_{k,\mathsf{G},\mathsf{B}} = 0$, $\beta = 0$, . . .
  - ▷ multi-class ON/OFF channels ($\mu_{k,B} = 0$)
  - ▷ maximal stability and fluid-optimality in i.i.d. case

Real Wireless Data Networks

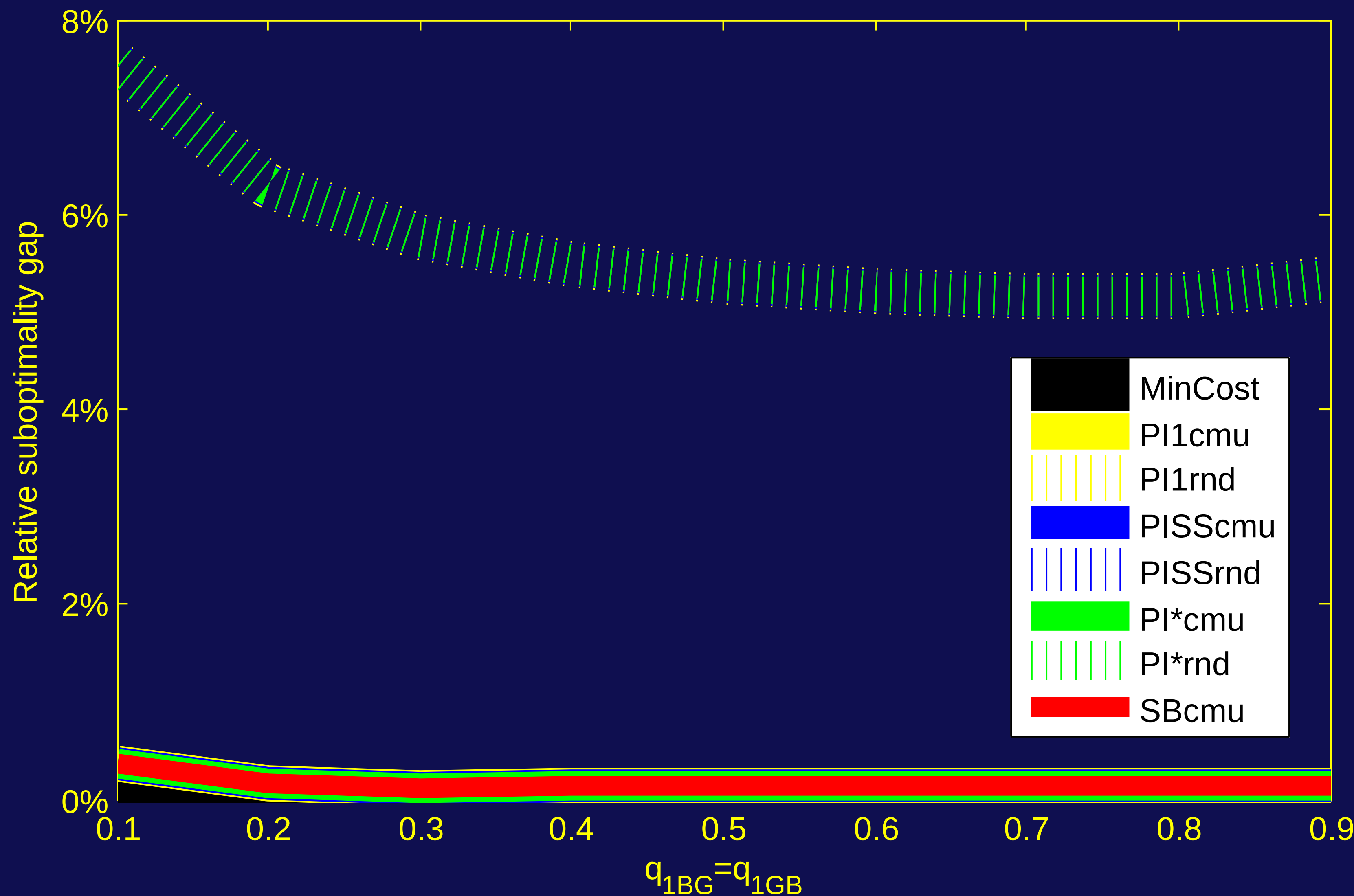
- $s_{k,n}$ and ε is usually known (E.g.: CDMA 1xEV-DO)
- PI* rule requires information of
  - ▷ expected job size $\mathbb{E}[B_k]$ (for both B, G)
  - ▷ state-transition matrix $\boldsymbol{Q}_k$ (for B)
- Approximations:
  - ▷ probability of departure $\mu_{k,n} \approx s_{k,n} \cdot \varepsilon / \mathbb{E}[B_k]$
  - ▷ for long jobs $q^*_{k,B,G} \approx q^{\mathsf{SS}}_{k,G}$
  - ▷ using both, index of B becomes independent of $\mathbb{E}[B_k]$
  - ▷ only tie-breaking of G jobs is $c_k s_{k,G} / \mathbb{E}[B_k]$

Systems with Random Arrivals

- We evaluate performance in experiments
  - ▷ $M = 1$
  - ▷ consider 2 different classes of jobs
  - ▷ $\lambda_{k,n}$: probability of arrival from class k to state n
- Schedulers: PI*, PI-SS, PI1
  - ▷ randomized and $c\mu$ tie-breaking in G
- Score Based (Bonald, 2004): $\nu_{k,n}^{\mathsf{SB}} := \sum_{m=1}^{n} q_{k,n,m}$
  - ▷ $c\mu$ tie-breaking in G

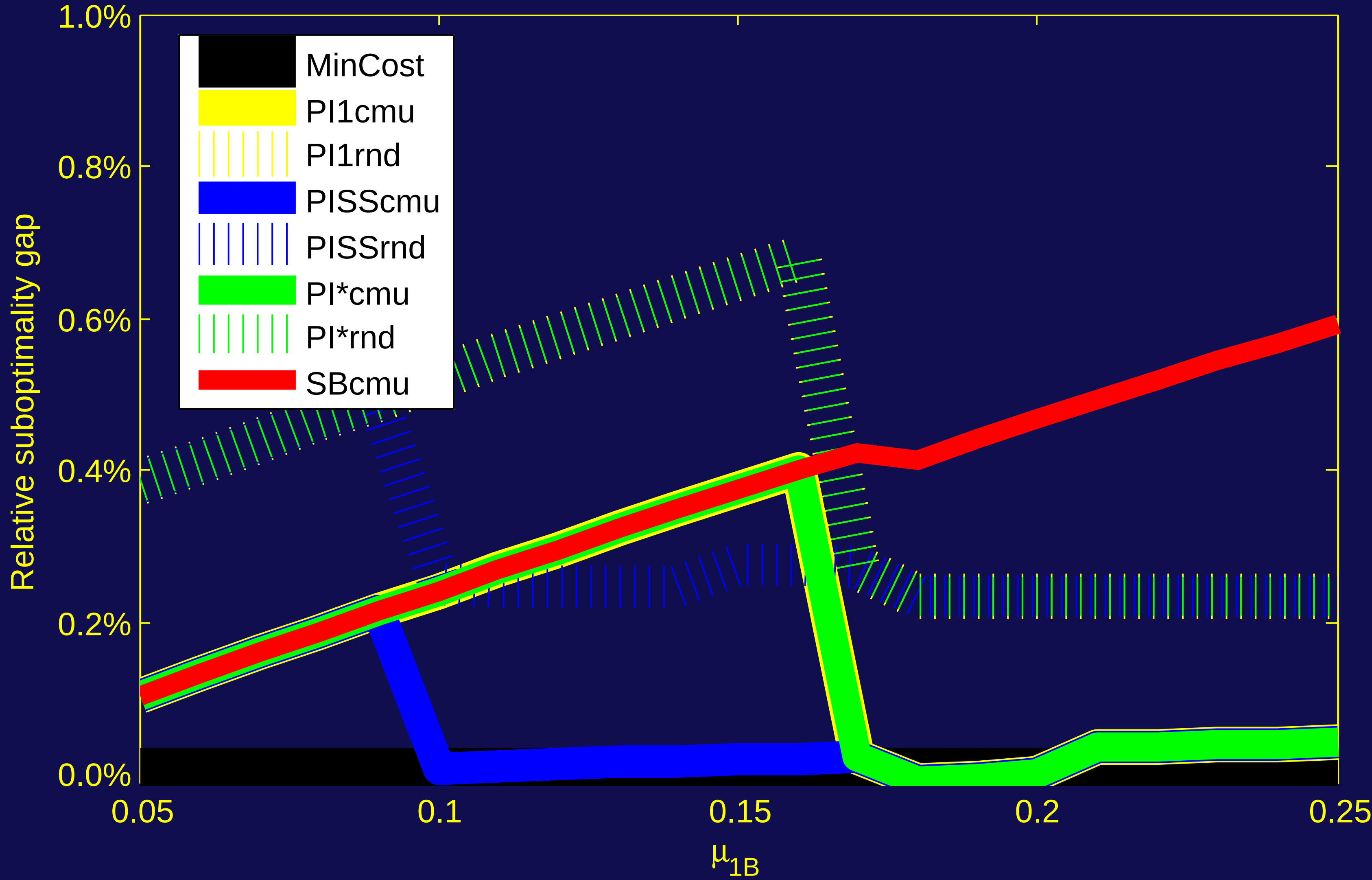
Experiments: Scenario 1

- Class 1 channel varies from slow-fading to fast-fading



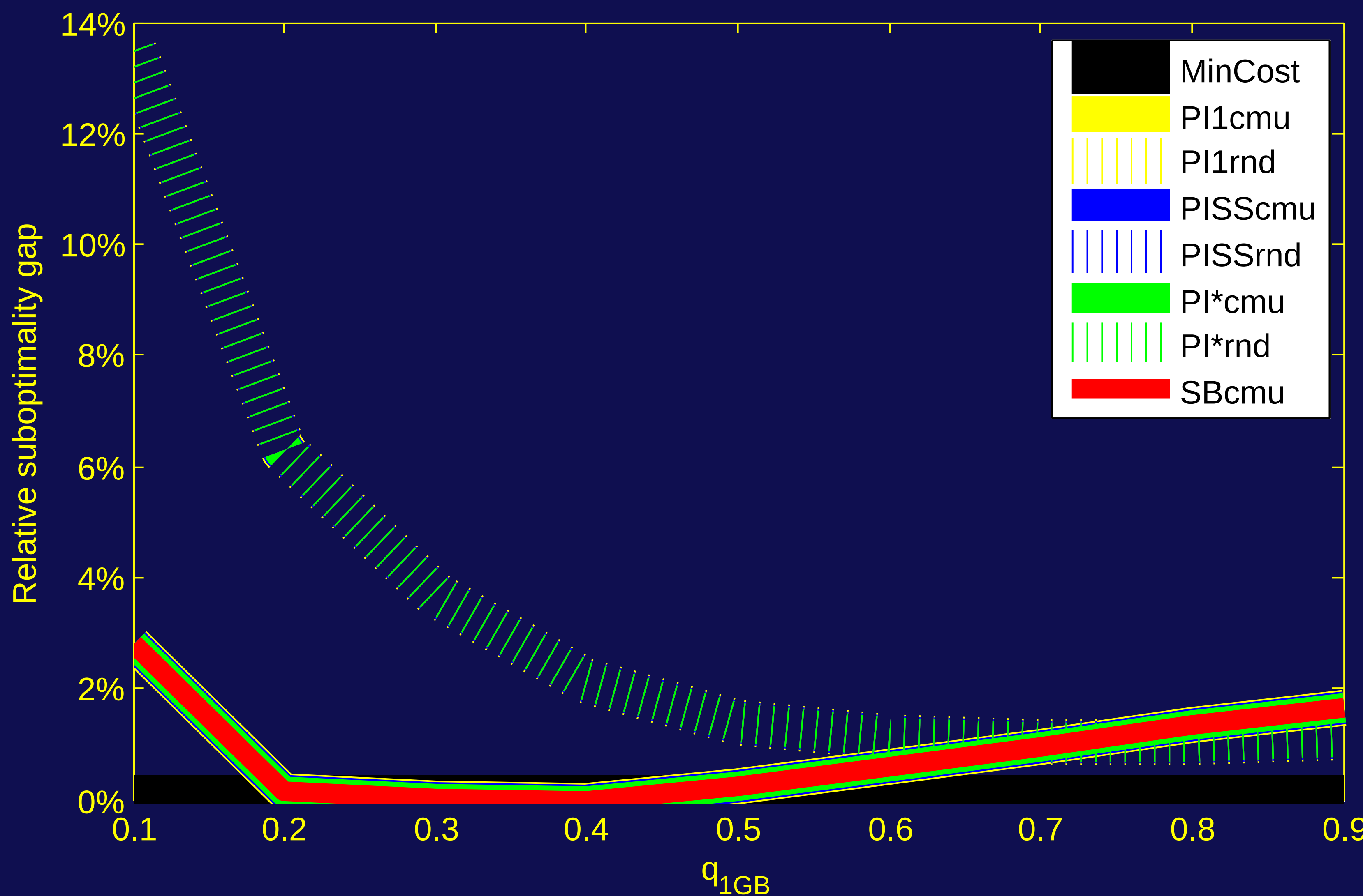
Experiments: Scenario 2

- Class 1: $\mu_{1,G} = 1$, $\mu_{1,B}$ varies



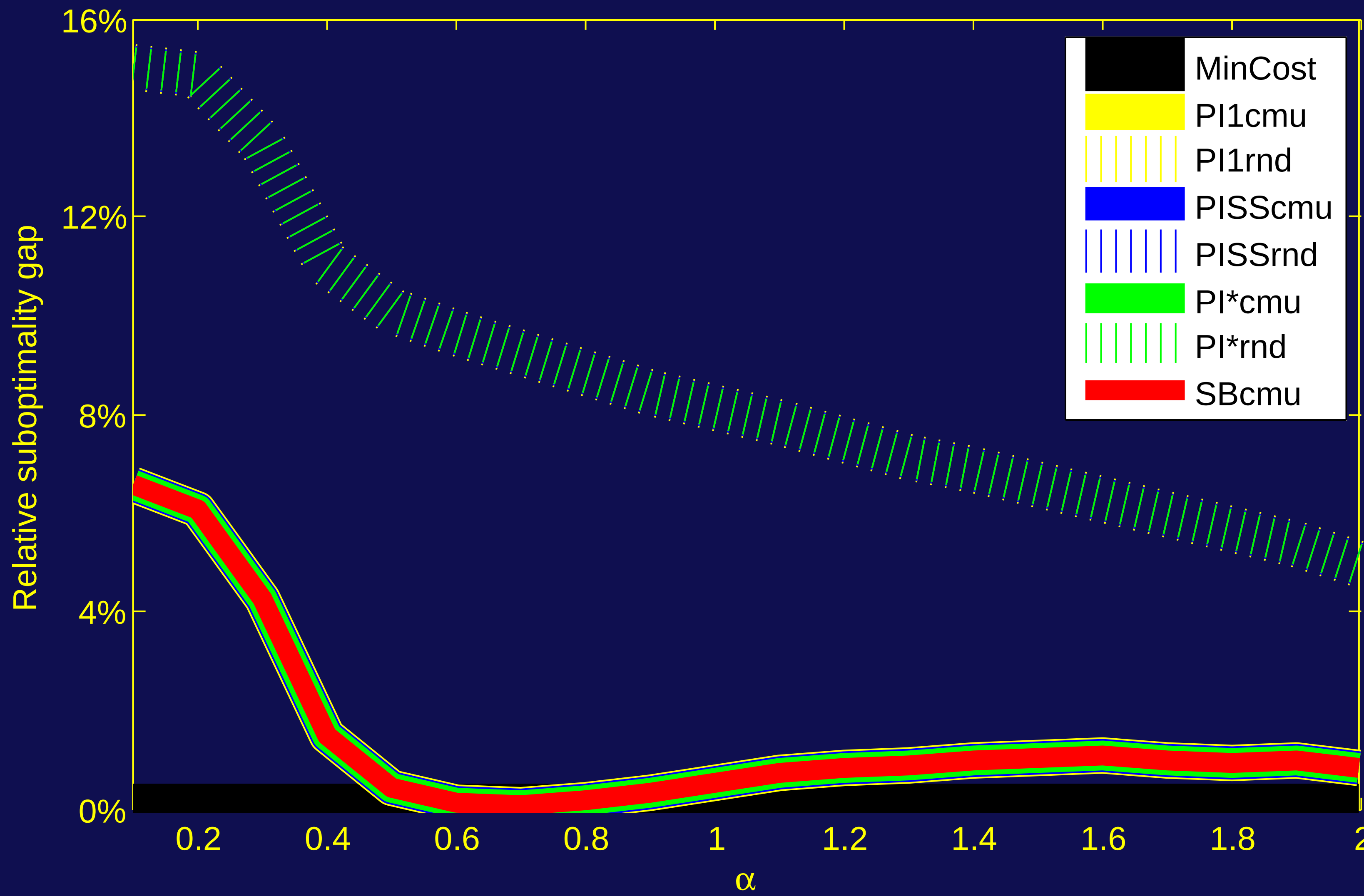
Experiments: Scenario 3

- Class 1: $q_{1,G,B}$ varies



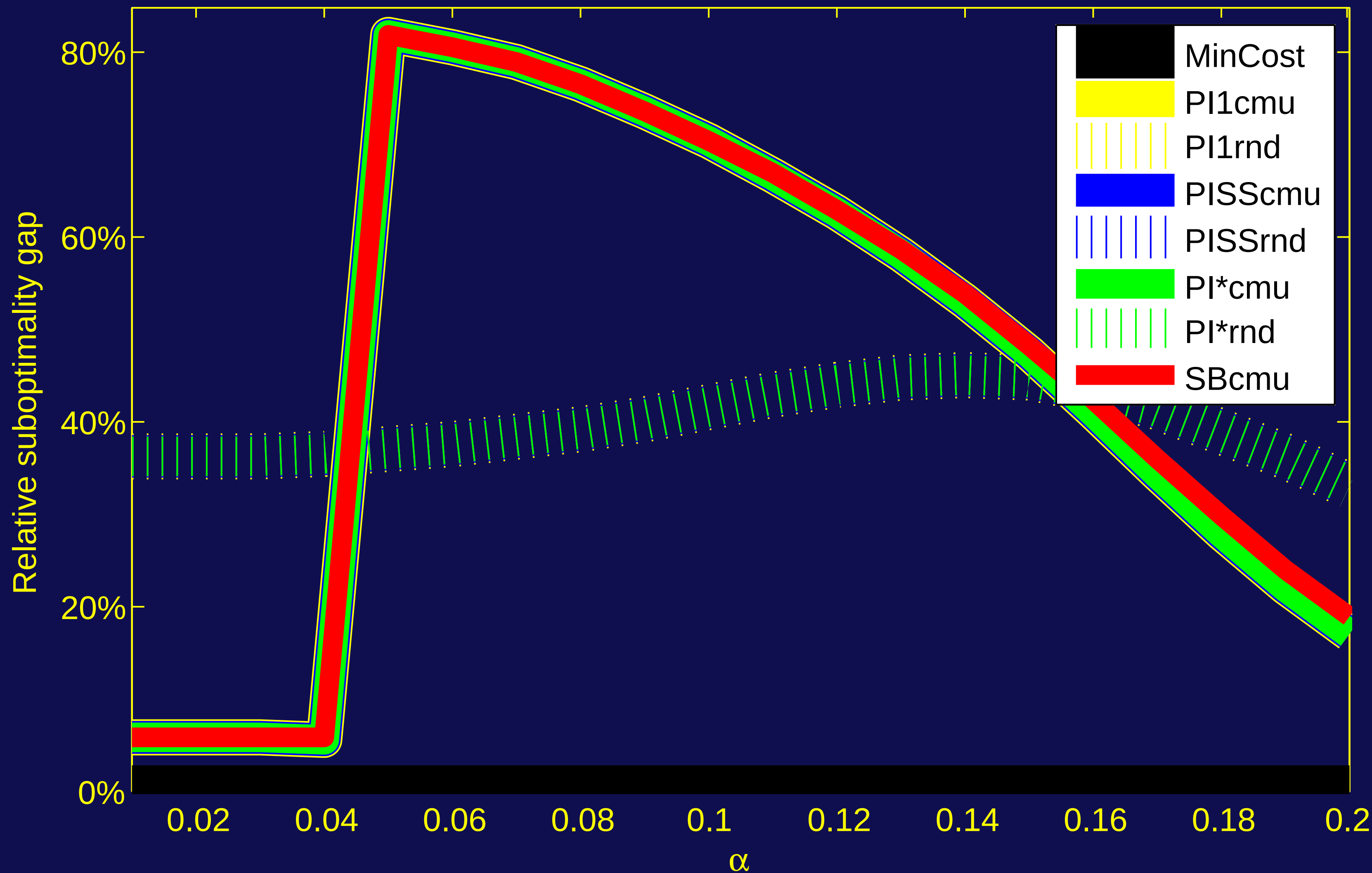
Experiments: Scenario 4

- Class 1: both $\mu_{1,G}$ and $\mu_{1,B}$ vary (decreasing job size)



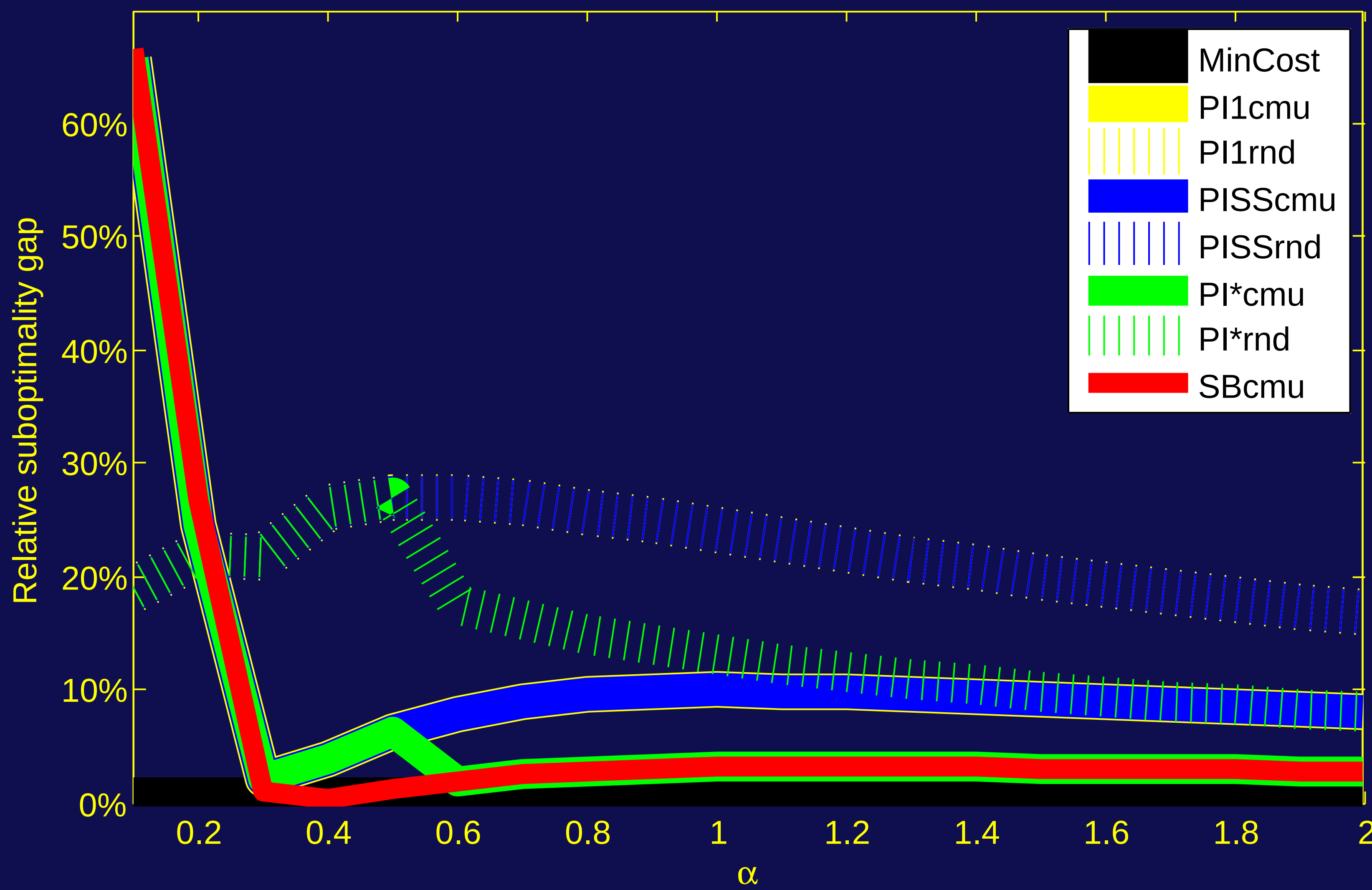
Experiments: Scenario 5

- Class 2: both $\mu_{2,G}$ and $\mu_{2,B}$ vary (decreasing job size)



Experiments: Scenario 6

- Class 2: both $\mu_{2,G}$ and $\mu_{2,B}$ vary (decreasing job size)



Experiments Summary

- PI variants are often nearly-optimal
- Tie-breaking in G more important than what is done in B
- $c\mu$ tie-breaking often significantly better than randomized
- The stability region seems similar to i.i.d. case

Conclusion

- New PI-like opportunistic rule
- Insights about value of information
- Open problems
  - ▷ PI* maximally-stable?
  - ▷ optimal solution (structure)
  - ▷ indices for more than 2-state channels (PI-like?)
  - ▷ general job sizes
  - ▷ partially observable channel conditions
  - ▷ correlation among users' channels

Thank you for your attention

Dynamic Prices (Index Values)

- We will assign a dynamic price to each user
- Arises in the solution of the parametric subproblem
  - ▷ optimal policy: use server iff price greater than ν
- Prices are values of ν when optimal solution changes
- However, such prices may not exist!
  - ▷ indexability has to be proved
- Price computation (if they exist):
  - ▷ in general, by parametric simplex method
  - ▷ by analysis sometimes obtained in a closed form

Optimal Solution to Subproblems

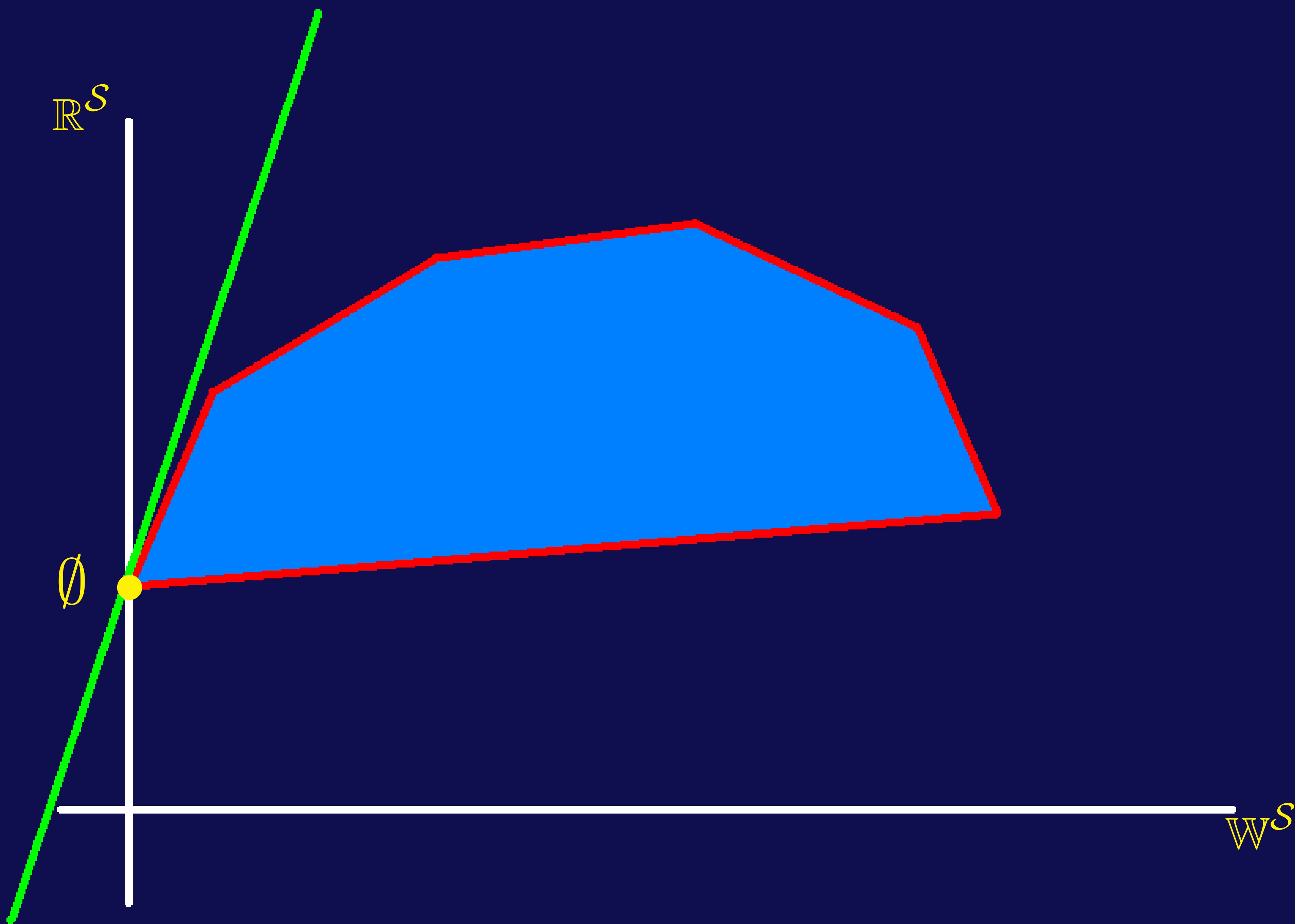
- For finite-state finite-action MDPs there exists an optimal policy that is deterministic, stationary, and independent of the initial state
  - ▷ we narrow our focus to those policies
  - ▷ represent them via serving sets $\mathcal{S} \subseteq \mathcal{N}$
  - ▷ policy $\mathcal{S}$ prescribes to serve in states in $\mathcal{S}$ and wait in states in $\mathcal{S}^{\mathsf{C}} := \mathcal{N} \setminus \mathcal{S}$
- Combinatorial ν-cost problem: $\max_{\mathcal{S} \subseteq \mathcal{N}} \mathbb{R}_n^{\mathcal{S}} - \nu \mathbb{W}_n^{\mathcal{S}}$, where

$$\mathbb{R}_n^{\mathcal{S}} := \mathbb{E}_n^{\mathcal{S}} \left[\sum_{t=0}^{\infty} \beta^t R_{X(t)}^{a(t)} \right], \quad \mathbb{W}_n^{\mathcal{S}} := \mathbb{E}_n^{\mathcal{S}} \left[\sum_{t=0}^{\infty} \beta^t W_{X(t)}^{a(t)} \right]$$

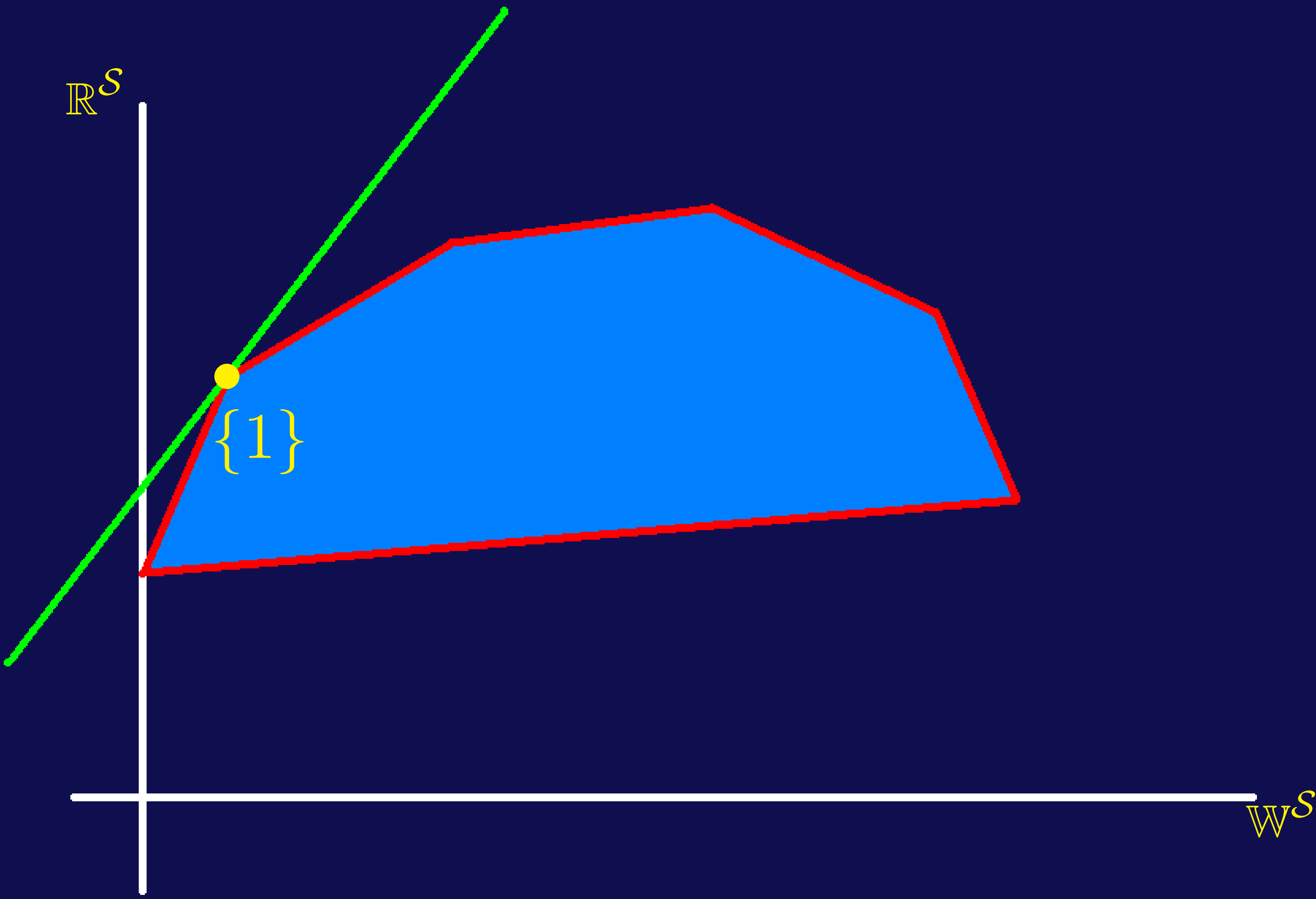
Geometric Interpretation

- $(\mathbb{W}_n^{\mathcal{S}}, \mathbb{R}_n^{\mathcal{S}})$ gives rise to 2-dim. performance region
- Indexability means the performance region is convex
- Optimal (threshold) policies are extreme points of the upper boundary of the performance region
- Index values are slopes of the upper boundary
- Indexability is sort of a dual concept to threshold policies
  - ▷ but not equivalent!

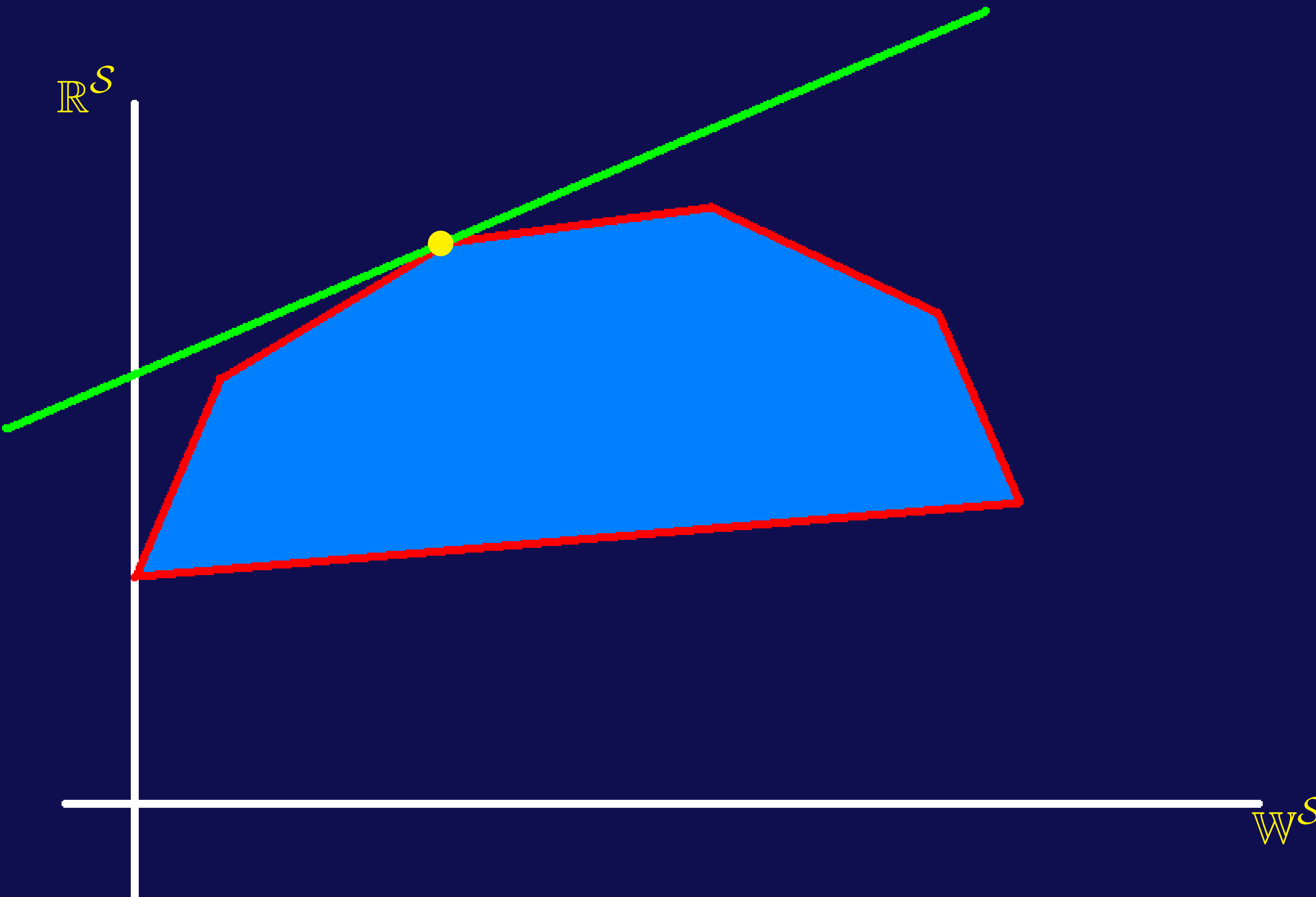
Performance Region



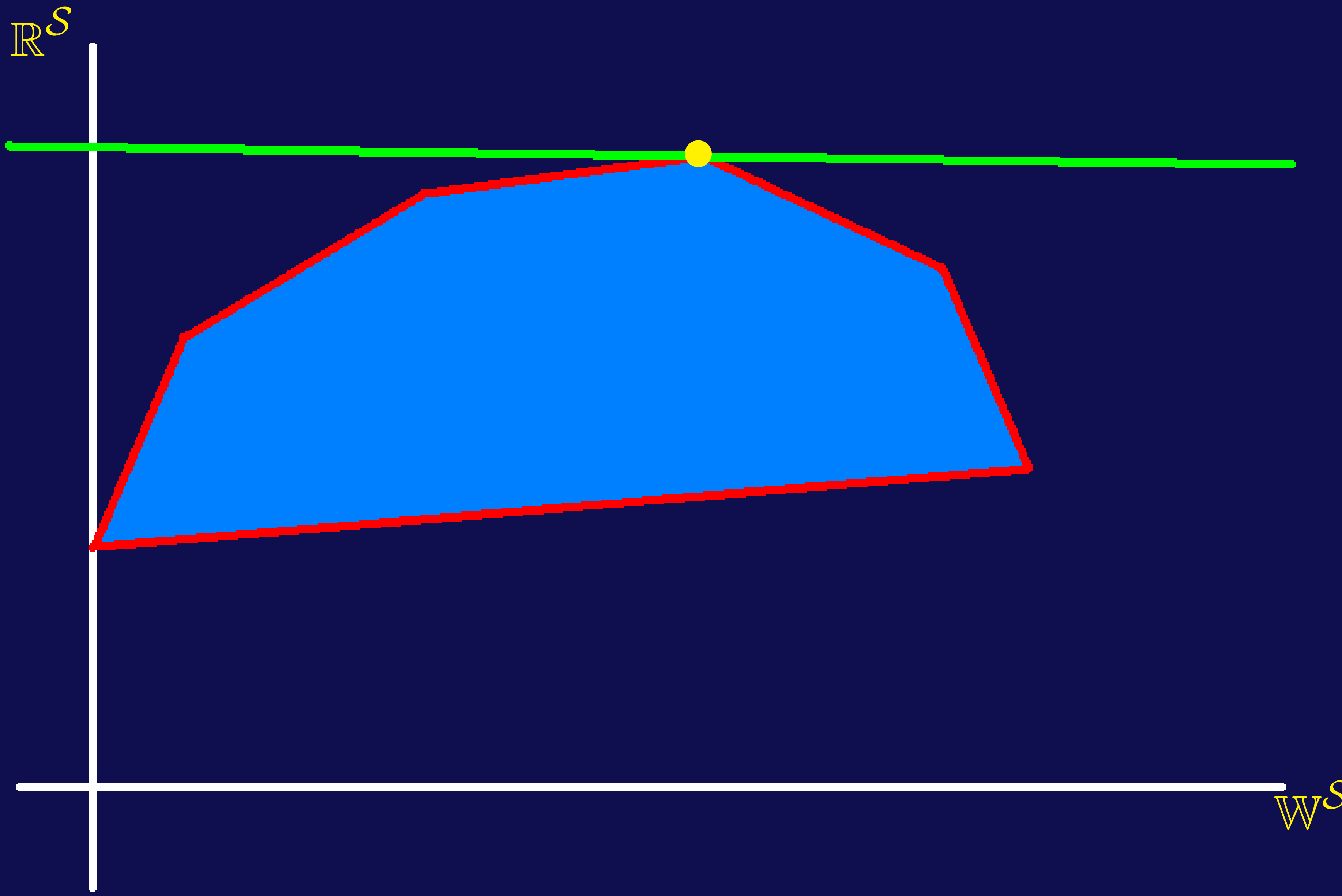
Performance Region



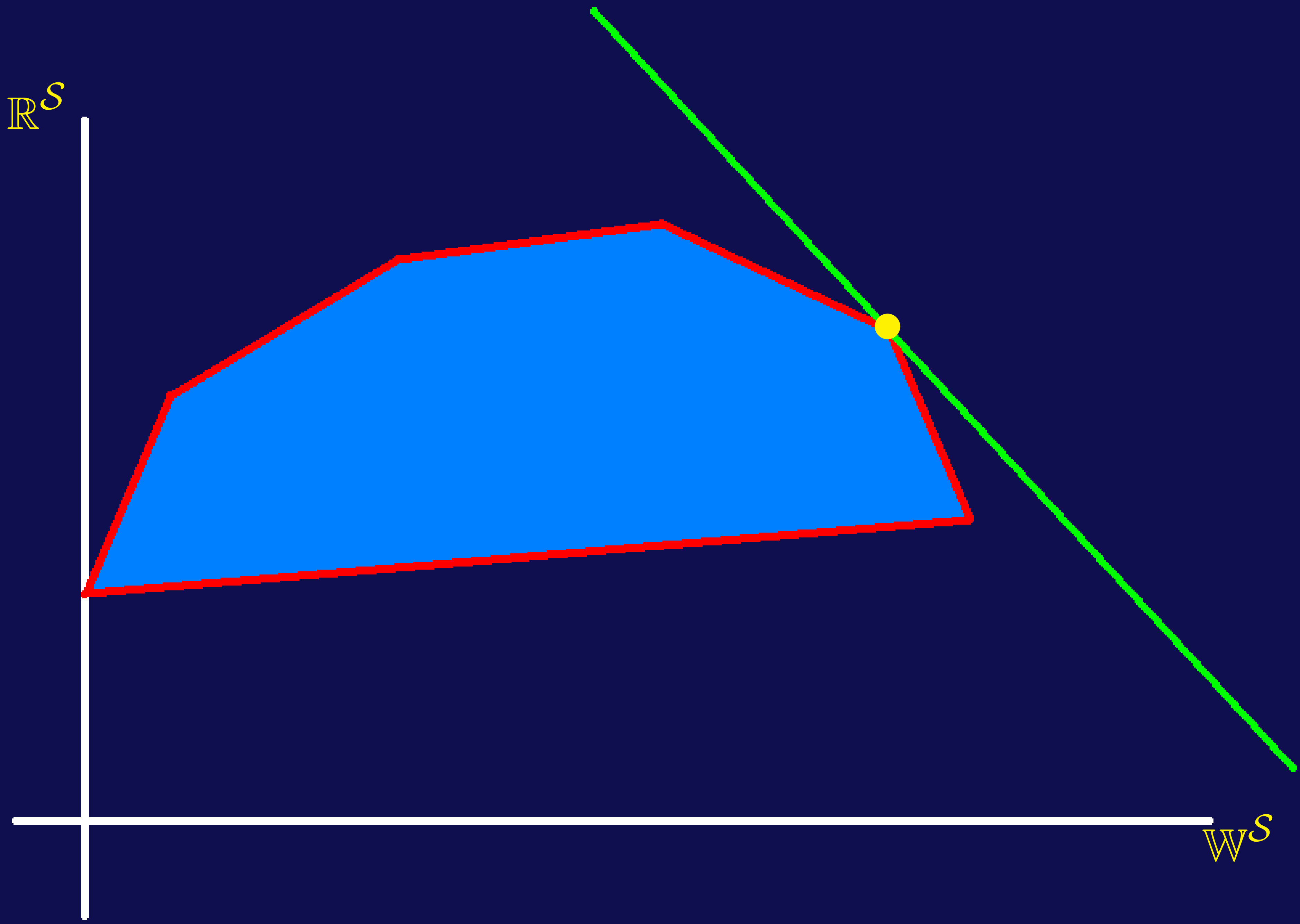
Performance Region



Performance Region



Performance Region



Performance Region