Opportunistic Schedulers for Optimal Scheduling of Flows in Wireless Systems with ARQ Feedback

Peter Jacko*, Sofía S. Villar†

ITC24 Krakow, September 5, 2012

*BCAM — Basque Center for Applied Mathematics, Spain
†UC3M — Universidad Carlos III de Madrid, Spain

BCAM, Bilbao

Motivation: Wireless Downlink

- CDMA 1xEV-DO, LTE
- Channel conditions vary randomly due to fading
- Channel conditions independent across users
- No interference
- Base station can serve M users per slot

Our Scheduler (Homogeneous Users)

- Which users to serve at every slot?
- After modelling and some math...
- Three FIFO priority lists of uncompleted flows:

1. high priority: all the users served in the previous slot whose feedback gave good condition
2. medium priority: all the users with no known feedback
3. low priority: all the users whose last feedback gave bad condition

- (For heterogeneous users: non-FIFO ordering)

Talk Outline

- Existing models and schedulers
- Our POMDP model (real-state restless bandit)
- Our scheduler
- Experiments

Models

- Availability of Current State Information (CSI):
  - ▷ channel-aware: CSI
  - ▷ delayed: CSI from previous slot
  - ▷ ARQ feedback: CSI from the last slot when served
- Scheduling level:
  - ▷ packet-level: persistent users with queues
  - ▷ flow-level: arriving and departing users
- Channel state evolution:
  - ▷ iid: stationary evolution
  - ▷ Markovian: knowing the matrix vs steady-state

Models

- Channel states:
  - ▷ General: N states (CDMA: $N = 11$, LTE: $N = 16$)
  - ▷ Gilbert-Elliot: 2 states (good/bad)
- Users:
  - ▷ single-class: homogeneous
  - ▷ multi-class: heterogeneous
- This work:
  - ▷ flow-level, ARQ feedback, Markovian, Gilbert-Elliot, multi-class

History

- Scheduling with not-time-varying users
  - ▷ $c\mu$-rule is optimal (Smith '56, Buyukkoc et al. '85)
- MaxWeight scheduler (Tassiulas & Ephermides '93)
  - ▷ serving longest non-interfering queues
- Being opportunistic enhances capacity (Knopp & Humblet '95) — but is very unfair
- Proportionally Fair scheduler (Qualcomm CDMA standard, '00)
  - ▷ priority to highest: current rate / realized throughput

Flow-Level Iid Channel-Aware Schedulers

- Score Based (Bonald, '04):
  - ▷ priority to highest probability of not improving rate
- Proportionally Best (Aalto & Lassila, '10):
  - ▷ priority to highest: current rate / best rate
- Potential Improvement (Ayesta et al. '10):
  - ▷ to highest: current rate / potential rate improvement
  - ▷ tie-breaking in best state: shortest first
- Maximal stability, fluid optimality (Ayesta et al., '11)

Markovian-Channel Schedulers

- Myopic at packet-level (Zhao et al., '08)
  - ▷ staying-on-good round-robin
  - ▷ optimal for homogeneous users (ON/OFF channels)
- Potential Improvement at flow-level (Jacko '11):
  - ▷ to highest: current rate / potential rate improvement
  - ▷ tie-breaking in best state: shortest first
- ARQ-based at packet-level (Ouyang et al., '11):
  - ▷ involved formula: no interpretation
  - ▷ near-optimal for heterogeneous users (2-state channels)

Job Scheduling Problem

- Discrete time ($t = 0, 1, 2, \dots$), preemptive service
- Jobs $k = 1, 2, \dots$ 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 $0 \le s_{k,B} \le s_{k,G}$ bits per second
- Minimize total waiting cost while serving M jobs/slot

Observability

- Rate adaptation: $x \leq> \theta_k := \mu_{k,B}/\mu_{k,G}$
- If user k is scheduled in belief state x, then ARQ feedback:

$$o_{k,x} := \begin{cases} G, & \text{w. p. } (1-\mu_{k,B})x, \text{ if } x \leq \theta_k; \\ B, & \text{w. p. } (1-\mu_{k,B})(1-x), \text{ if } x \leq \theta_k; \\ *, & \text{w. p. } \mu_{k,B}, \text{ if } x \leq \theta_k; \\ G, & \text{w. p. } (1-\mu_{k,G})x, \text{ if } x > \theta_k; \\ B, & \text{w. p. } (1-x), \text{ if } x > \theta_k; \\ *, & \text{w. p. } x \cdot \mu_{k,G}, \text{ if } x > \theta_k; \end{cases}$$

POMDP 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 $\mathcal{N}_k := \{*\} \cup [0, 1]$
  - ▷ expected one-period capacity consumption $\boldsymbol{W}_k^a := a$
  - ▷ Expected one-period reward

$$R_{k,0}^1 := 0, \qquad R_{k,n}^1 := -c_k \cdot (1 - \mathbb{P}[o_{k,x} = *]),$$
$$R_{k,0}^0 := 0, \qquad R_{k,n}^0 := -c_k;$$

POMDP Model

- State process $N_k(t) \in \mathcal{N}_k$ transitions

$$N_k(t+1) = \begin{cases} N_k(t) q_{k,G,G} + (1 - N_k(t)) q_{k,B,G}, \\ \qquad\qquad \text{w.p. } 1, \text{ if } a_k(t) = 0; \\ q_{k,G,G}, \quad \text{w.p. } \mathbb{P}[o_{k,x} = G], \text{ if } a_k(t) = 1; \\ q_{k,B,G}, \quad \text{w.p. } \mathbb{P}[o_{k,x} = B], \text{ if } a_k(t) = 1; \\ *, \quad \text{w.p. } \mathbb{P}[o_{k,x} = *], \text{ if } a_k(t) = 1; \end{cases}$$

- Action process $a_k(t) \in \mathcal{A}$ – to be decided

Optimization Problem

- Formulation under the time-average criterion:

$$\max_{\pi \in \Pi} \lim_{T \to \infty} \frac{1}{T} \sum_{k \in \mathcal{K}} \mathbb{E}^{\pi} \left[\sum_{t=0}^{T-1} R_{k, X_k(t)}^{a_k(t)} \right]$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \mathbb{E}^{\pi} \left[W_{k, X_k(t)}^{a_k(t)} \right] = M, \ \text{for all } t = 0, 1, 2, \ldots$$

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

Relaxations

- Whittle ('88)
  - ▷ Serve M jobs on time-average
- Lagrangian
  - ▷ Pay cost ν for using the server
- Decomposes due to user independence into single-user parametric subproblems
- Provides an upper bound for RAP

Optimal Solution to Subproblems

- Conjecture: The problem is indexable, which implies
  - ▷ if $\nu \leq \nu_{k,x}$, then it is optimal to serve in state n
  - ▷ if $\nu \geq \nu_{k,x}$, then it it optimal to wait in state n
- $\nu_{k,x}$ is the dynamic shadow price (Whittle index value)
- Proposition: Index $\nu_{k,x}$ is increasing in x
- Proposition: Threshold policy is optimal
  - ▷ serve iff user-k state is above a threshold
- Proposition: Index $\nu_{k,x} = +\infty$ if last feedback was G or no information at all

Optimal Solution to Subproblems

- Proposition: If last feedback was B, index is

$$\nu_{k,x} = c_k \left\{ \phi^*_{k,G} \frac{1-(1-\mu_{k,G})q_{k,G,G}}{(1-\rho_k)(q^{\mathsf{SS}}_{k,G}-x)[1-(1-\mu_{k,G})\phi^*_{k,G}]} + (T_k(x)+1)\left[\frac{(1-\mu_{k,G})q_{k,G,G}}{1-(1-\mu_{k,G})\phi^*_{k,G}} - 1\right]\right\}$$

where

$$\phi^*_{k,G} := \frac{1}{\frac{\mu_{k,G}}{\phi_k^{T_k}(q_{k,B,G})} + \frac{1-\mu_{k,G}}{\phi^{\mathsf{SS}}_{k,G}}}$$

Our Scheduler

- Whittle-index based:
  - ▷ serve M jobs with highest actual index
  - ▷ "asymptotically optimal" (Weber & Weiss '90)
- Absolute priority: if last feedback was G or no info
  - ▷ (if too many new arrivals) ordering according to

$$\nu_{k,x}^{(2)} = \frac{c_k \mu_{k,G} x}{1 - (1 - \mu_{k,G})(q_{k,G,G} - x)},$$

- Rest: if last feedback was B
  - ▷ ordering according to $\nu_{k,x}$
  - ▷ increasing index implies FIFO ordering within class

Remarks

- Initial state: $x = q^{\mathsf{SS}}_{k,G}$
- Analogy in iid channel-aware systems
  - ▷ maximal stability if absolute priority to users in G
  - ▷ fluid optimality: shortest-first tie-breaking
- Optimality if $q_{k,B,G} = q_{k,G,G}$ for all k (classic $c\mu$-rule)
- Simplifies in single-class systems
  - ▷ equivalent to myopic scheduler $\nu^{\text{myopic}}_{k,x} := c_k \mu_{k,G} x$
  - ▷ equivalent to belief scheduler $\nu^{\text{belief}}_{k,x} := x$

Illustration 1

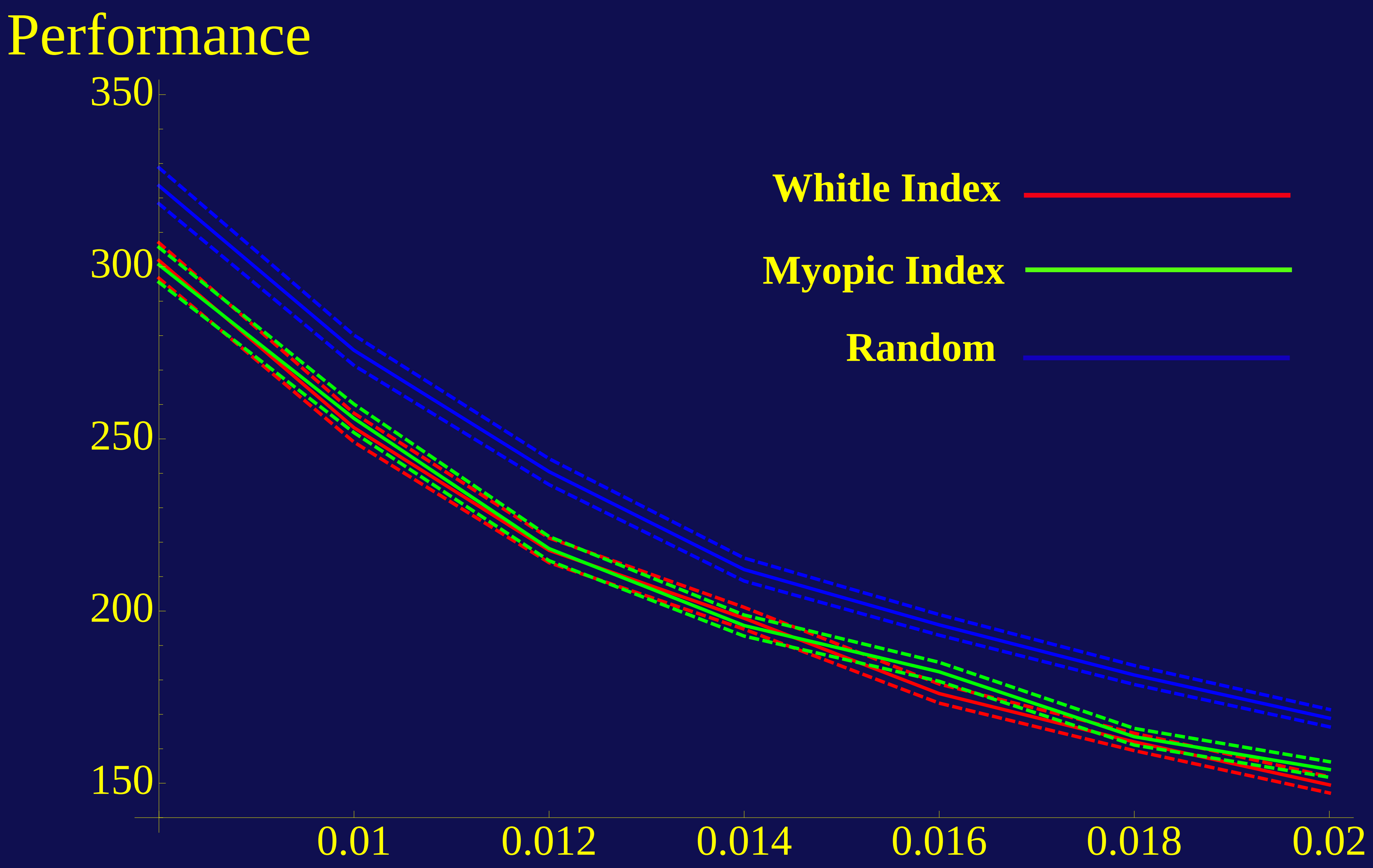
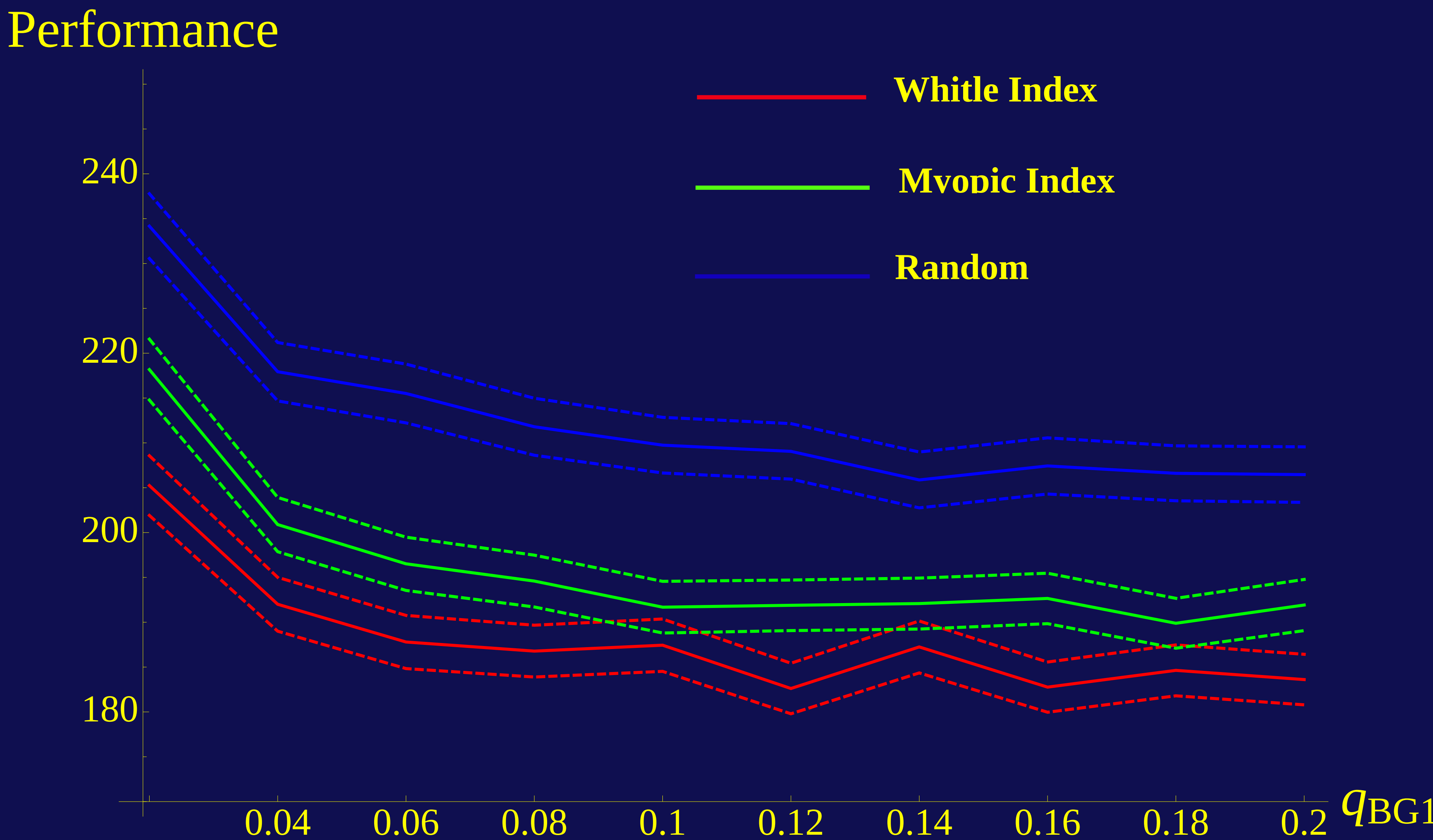


Illustration 2



Conclusion

- New practical scheduler, generalizes existing
- Introduced belief scheduler, often equivalent
- Should/could be done
  - ▷ testing
  - ▷ optimality in single-class systems
  - ▷ maximal stability
  - ▷ asymptotic optimality
- Open problems
  - ▷ extension to N states
  - ▷ general job sizes

Thank you for your attention

Other Scheduling Disciplines

- Relatively Best (Qualcomm CDMA standard, 2000):

$$\nu_{k,n}^{\mathsf{RB}} := \frac{\mu_{k,n}}{\sum_{m=1}^{N_k} q_{k,m}\mu_{k,m}}$$

  - ▷ ≈ Proportionally Fair scheduler (Borst, 2005)

- Score Based (Bonald, 2004): $\nu_{k,n}^{\mathsf{SB}} := \sum_{m=1}^{n} q_{k,m}$

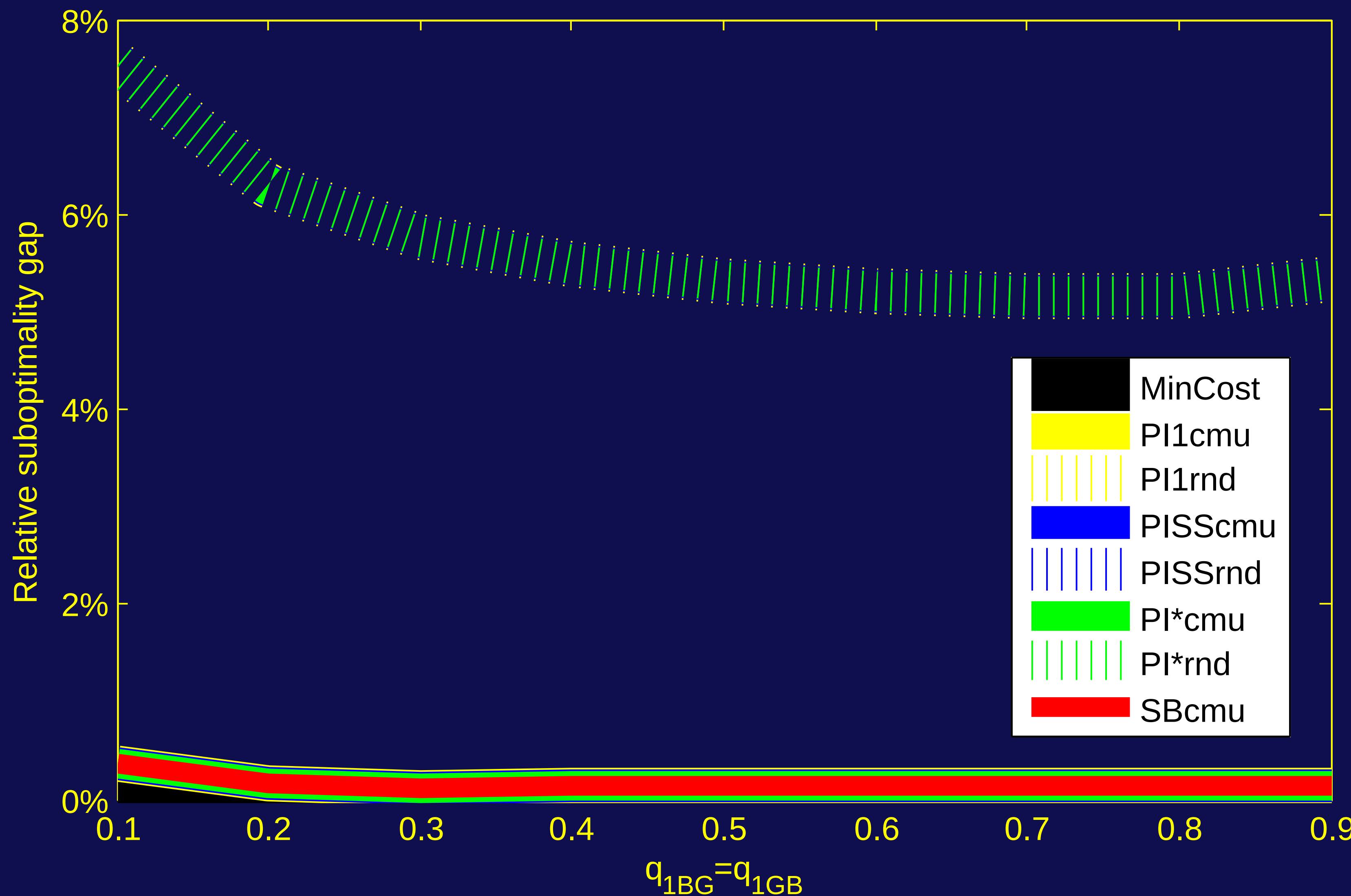
- Proportionally Best: $\nu_{k,n}^{\mathsf{PB}} = \dfrac{\mu_{k,n}}{\mu_{k,N_k}}$
  - ▷ maximum stability region (Aalto & Lassila, 2010)

Systems with Random Arrivals

- PI rule has maximum stability region
  - ▷ the only rule under general c_k's
- PI equivalent to RB in "symmetric" systems
  - ▷ performance characterized as processor sharing
- We evaluate performance in simulations
  - ▷ consider 2 different classes of jobs
  - ▷ λ_k: probability of arrival from class k

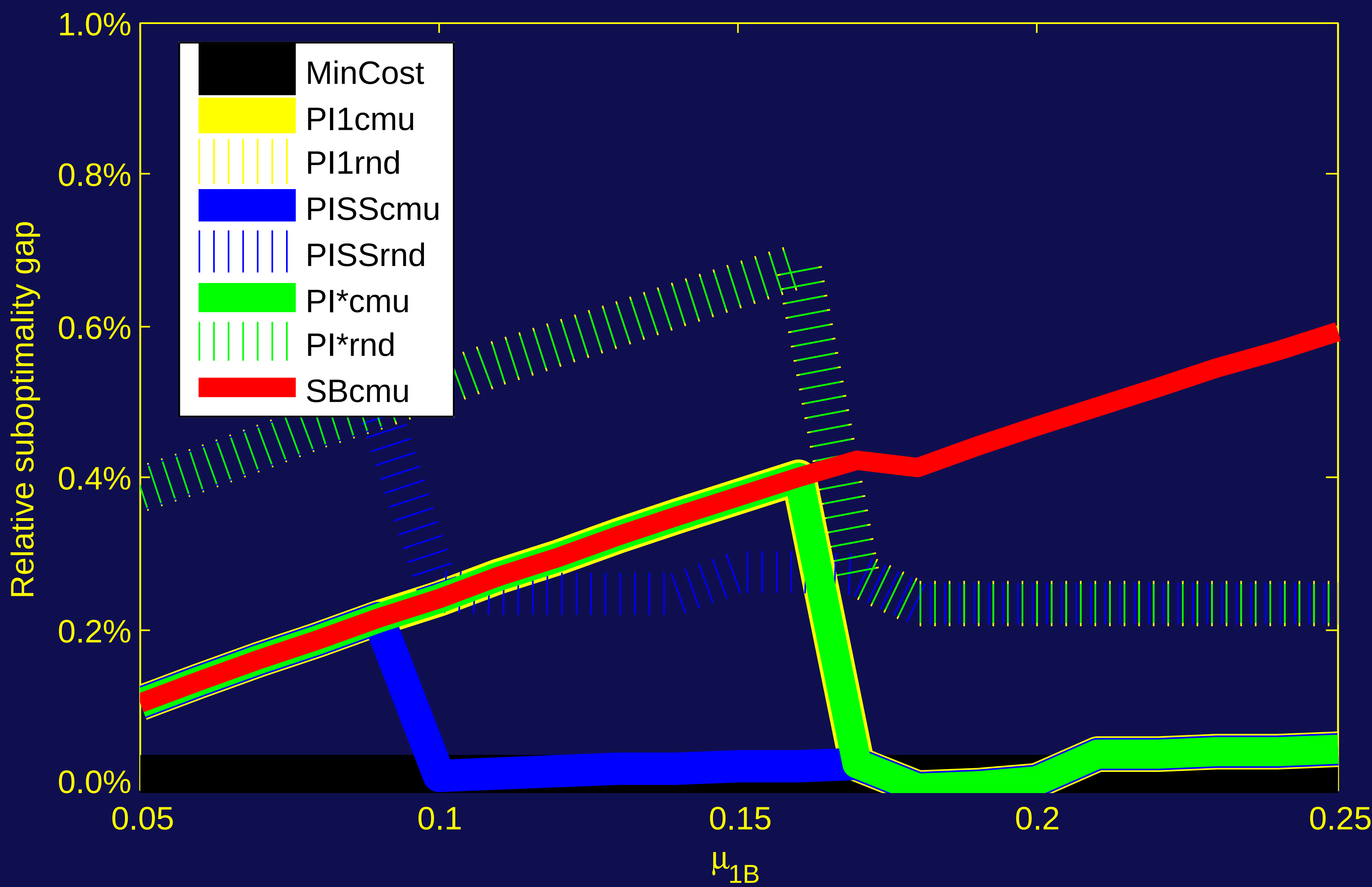
Experiments: Scenario 1

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



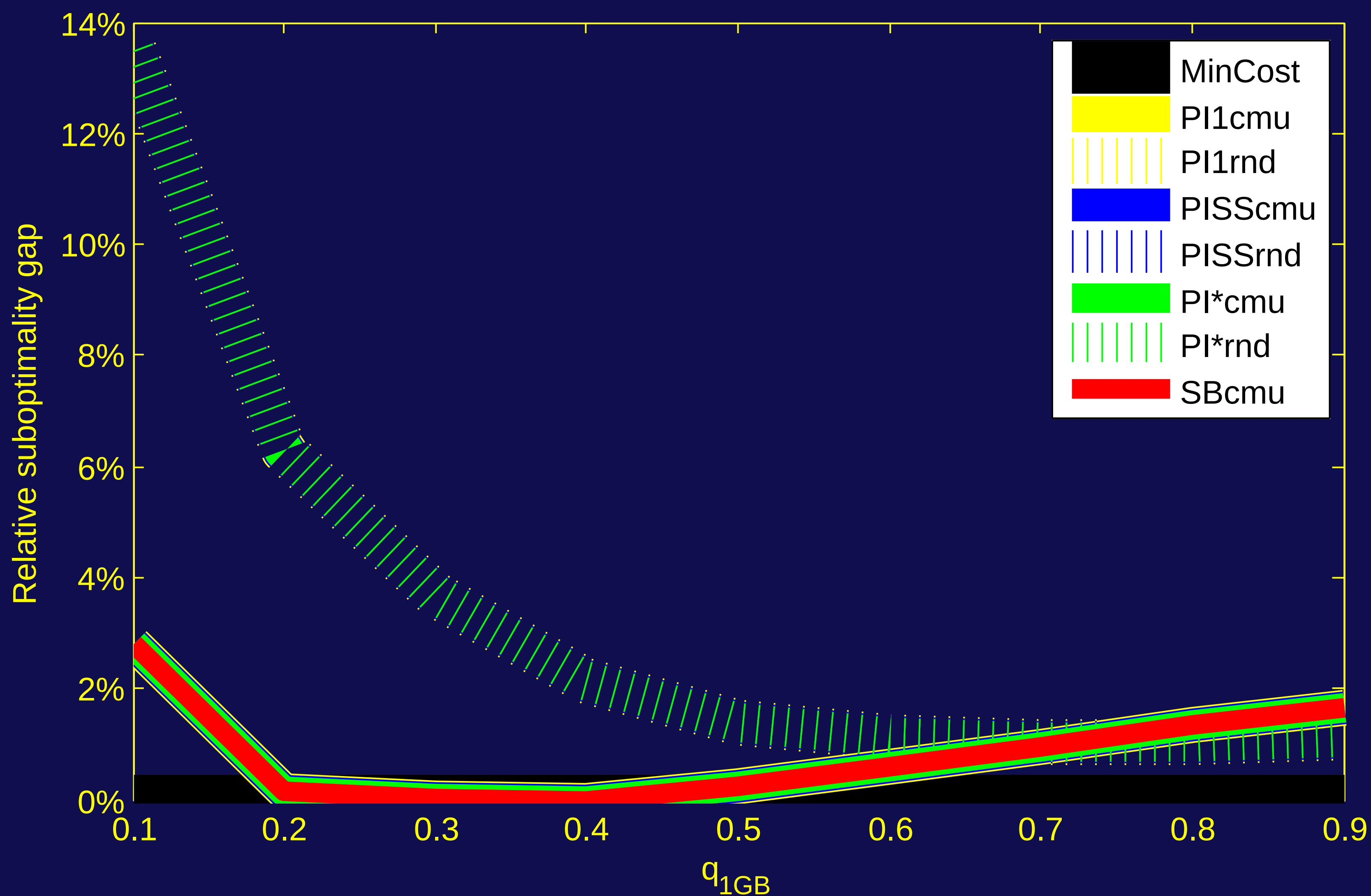
Experiments: Scenario 2

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



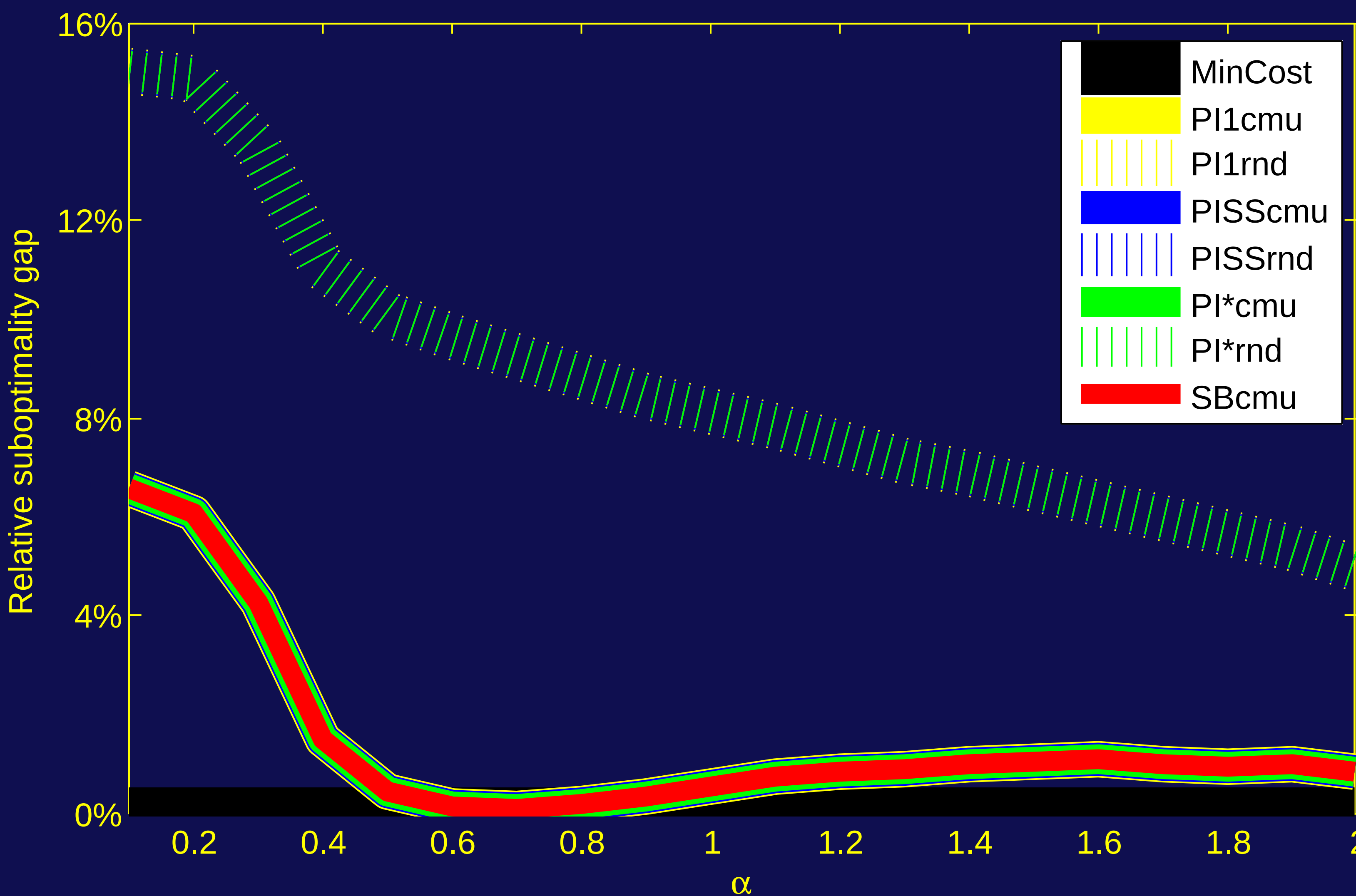
Experiments: Scenario 3

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



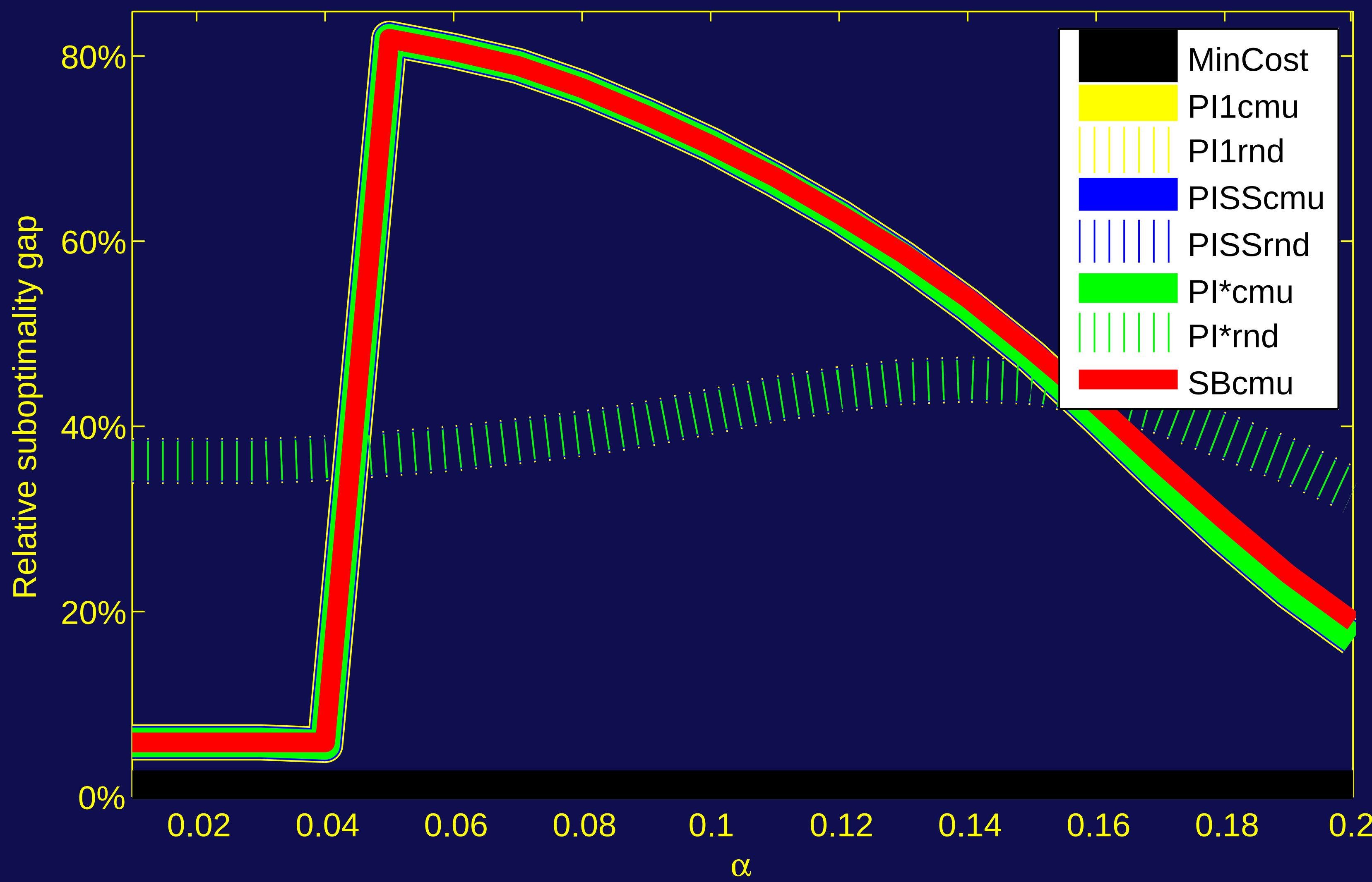
Experiments: Scenario 4

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



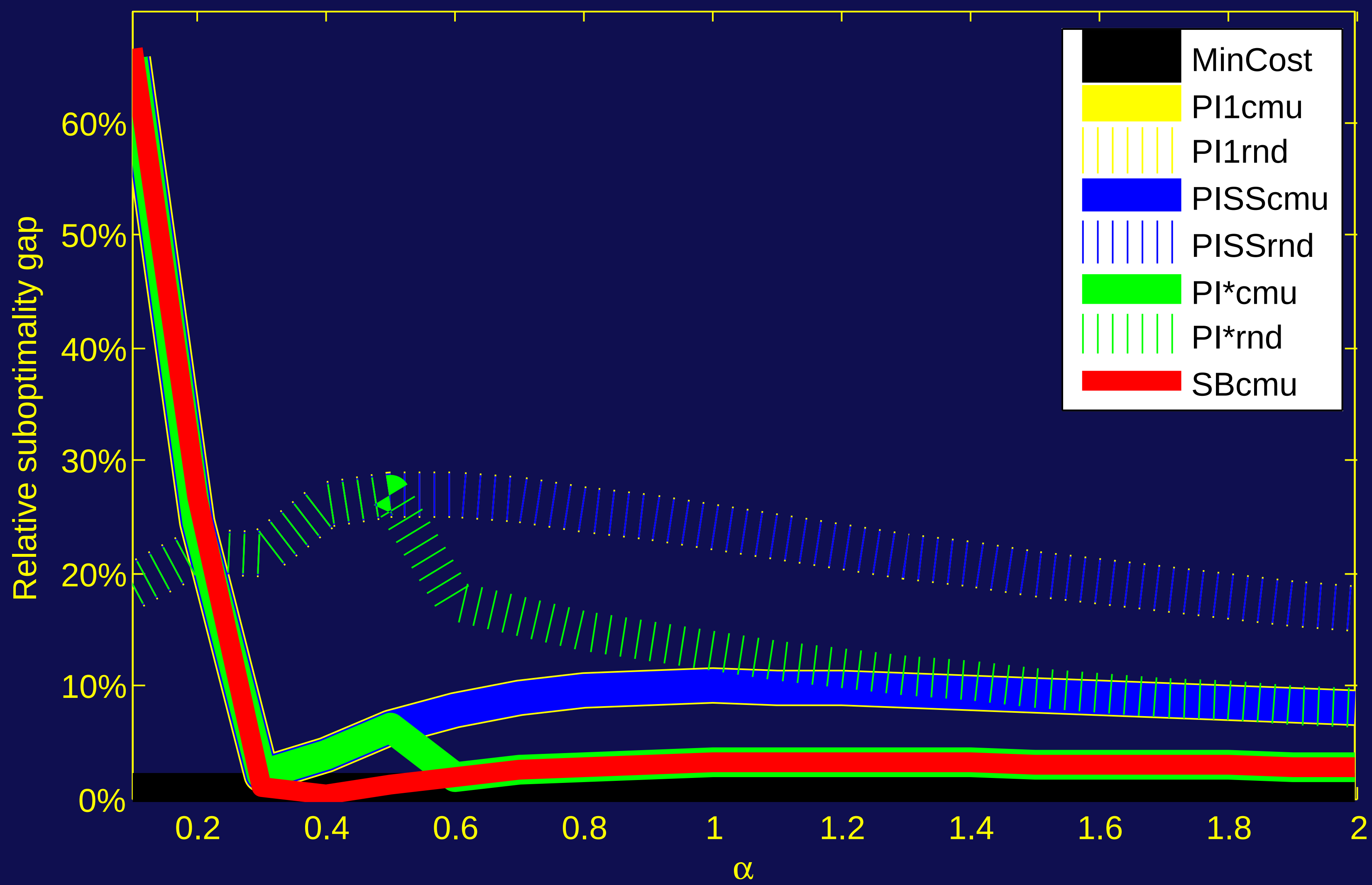
Experiments: Scenario 5

- 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)

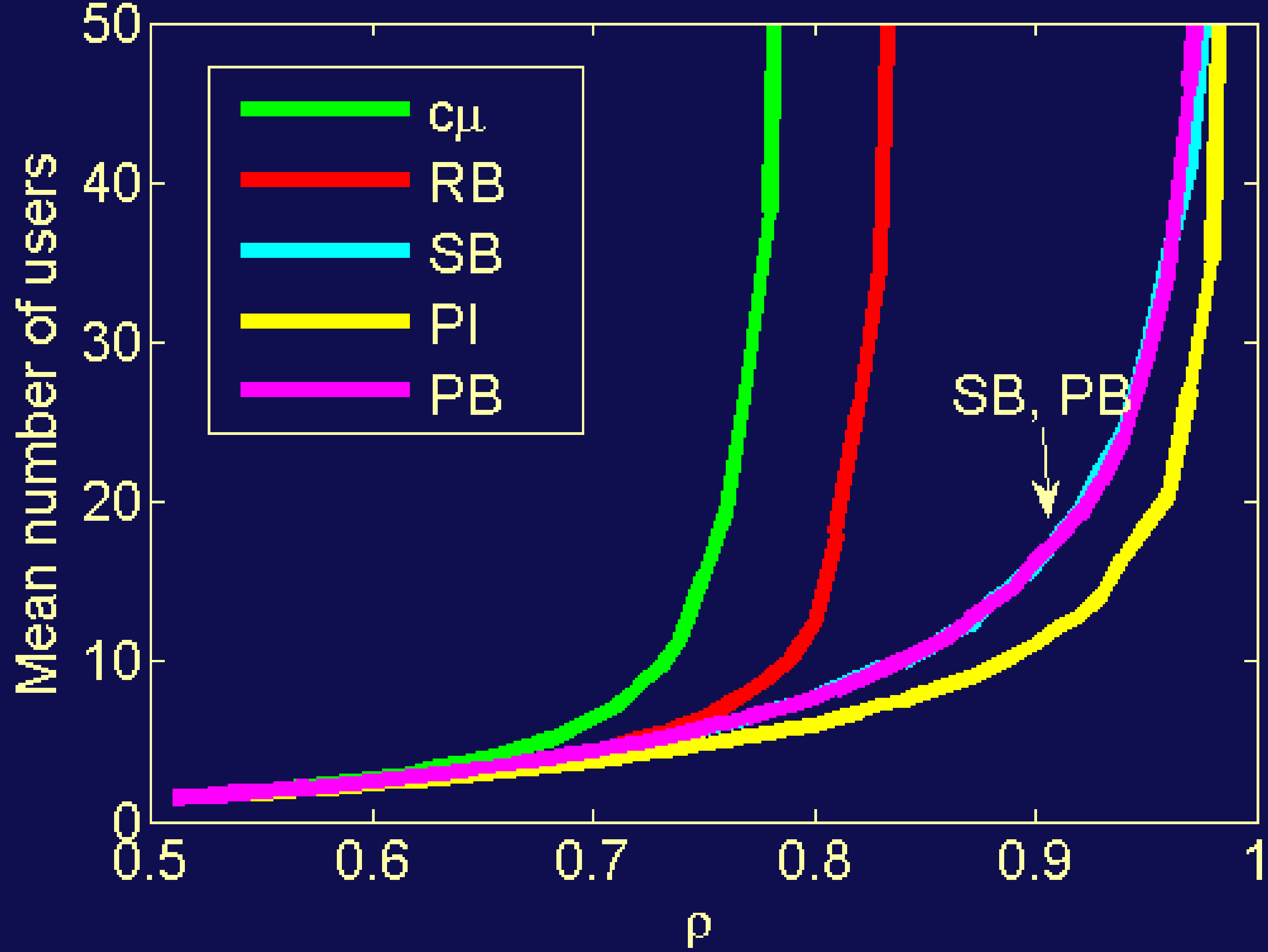


Experiments Summary

- 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

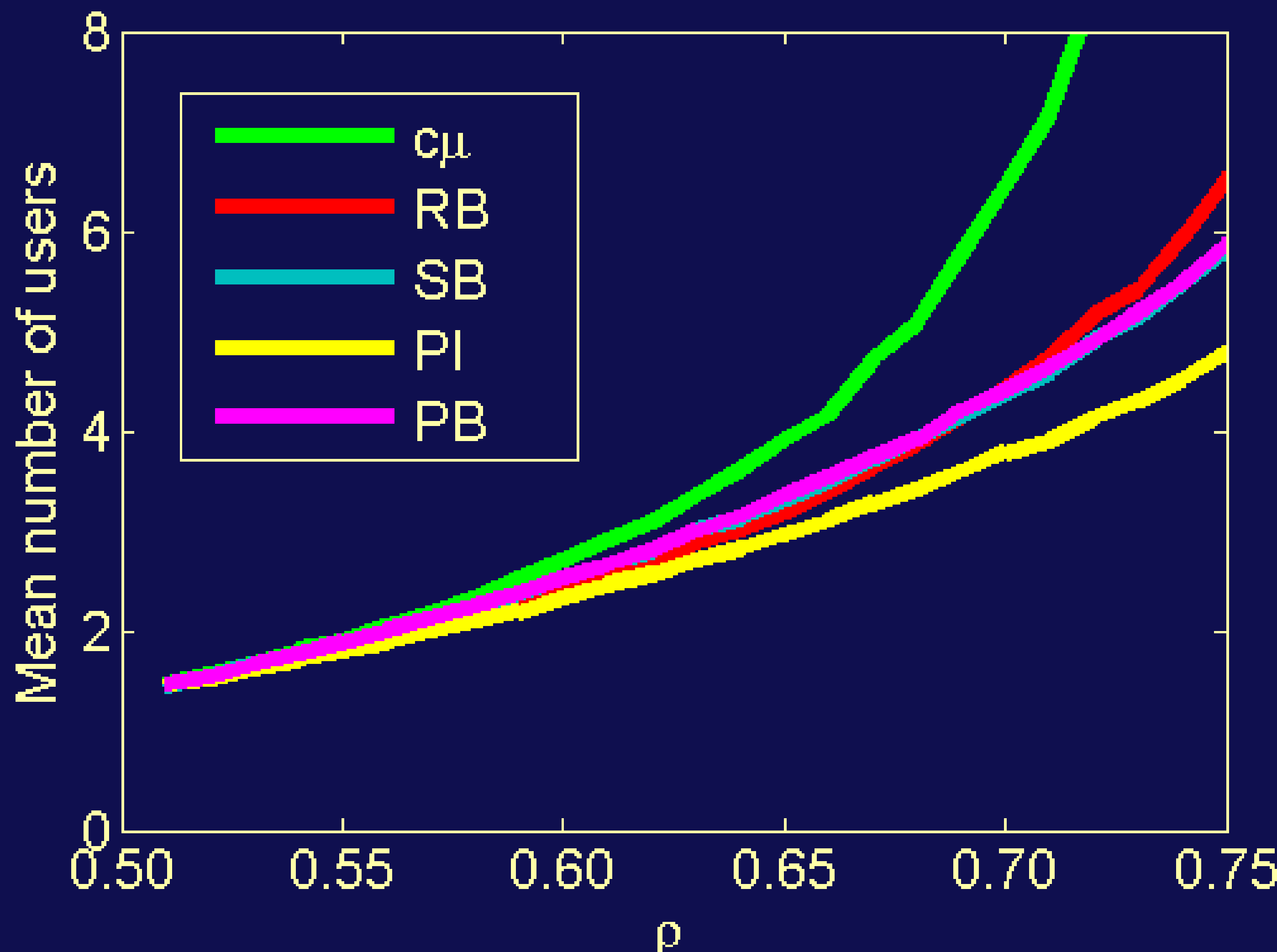
Numerical Simulations: Scenario 1

- Varied λ_1 so that ϱ varies from 0.5 to 1



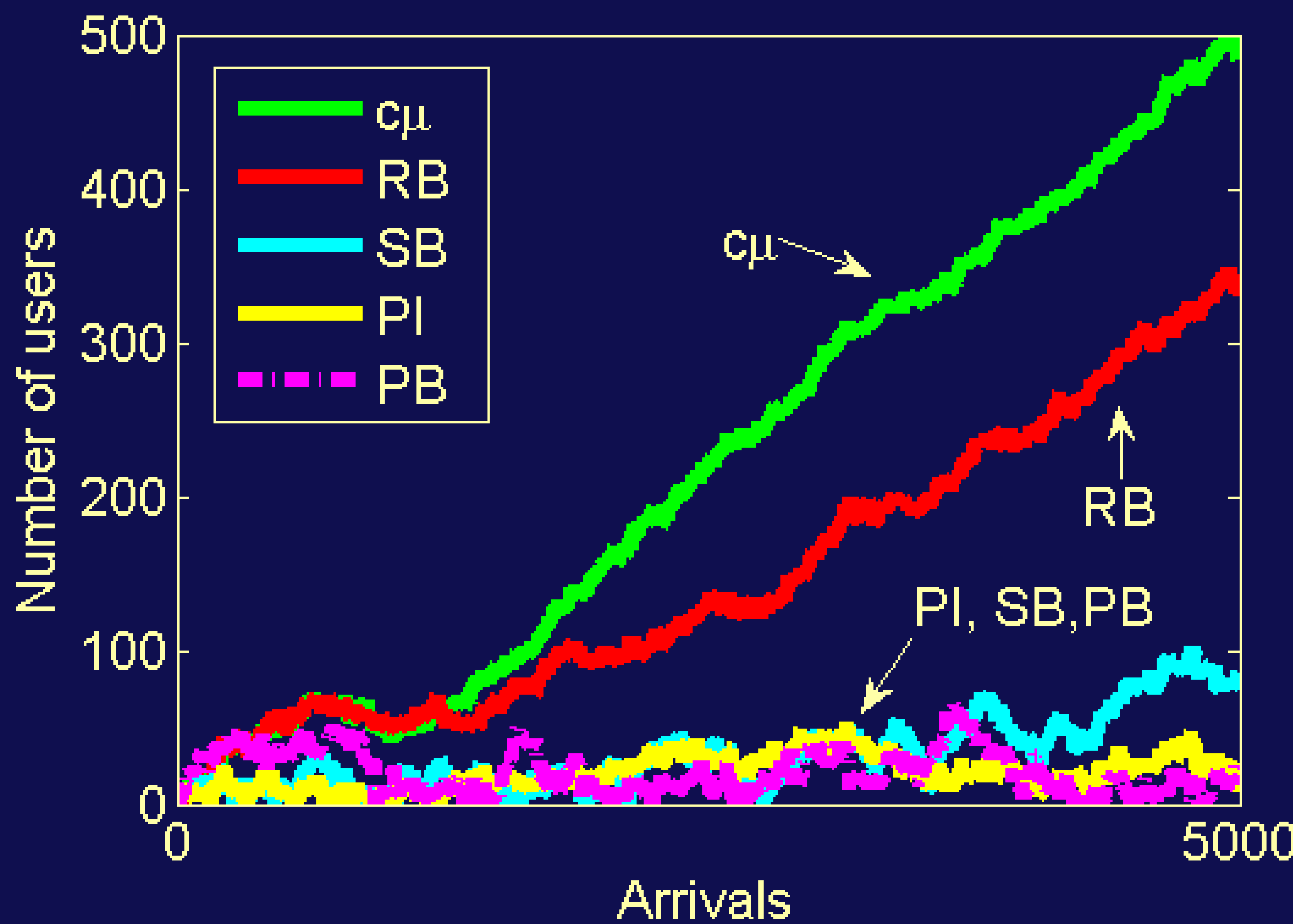
Numerical Simulations: Scenario 1

- Varied λ_1 so that ϱ varies from 0.5 to 1



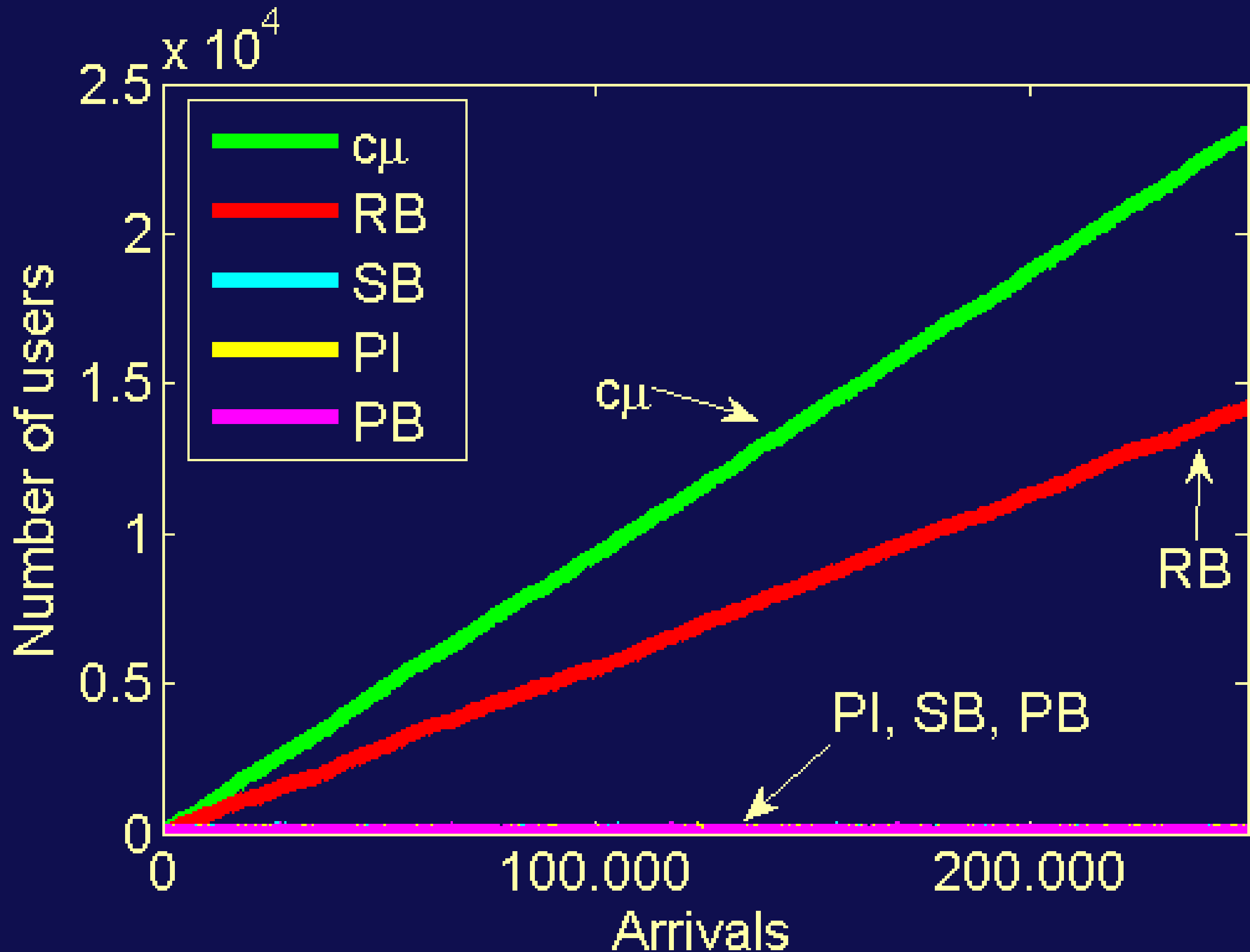
Numerical Simulations: Scenario 1

- Sample path of the number of users, $\varrho = 0.95$



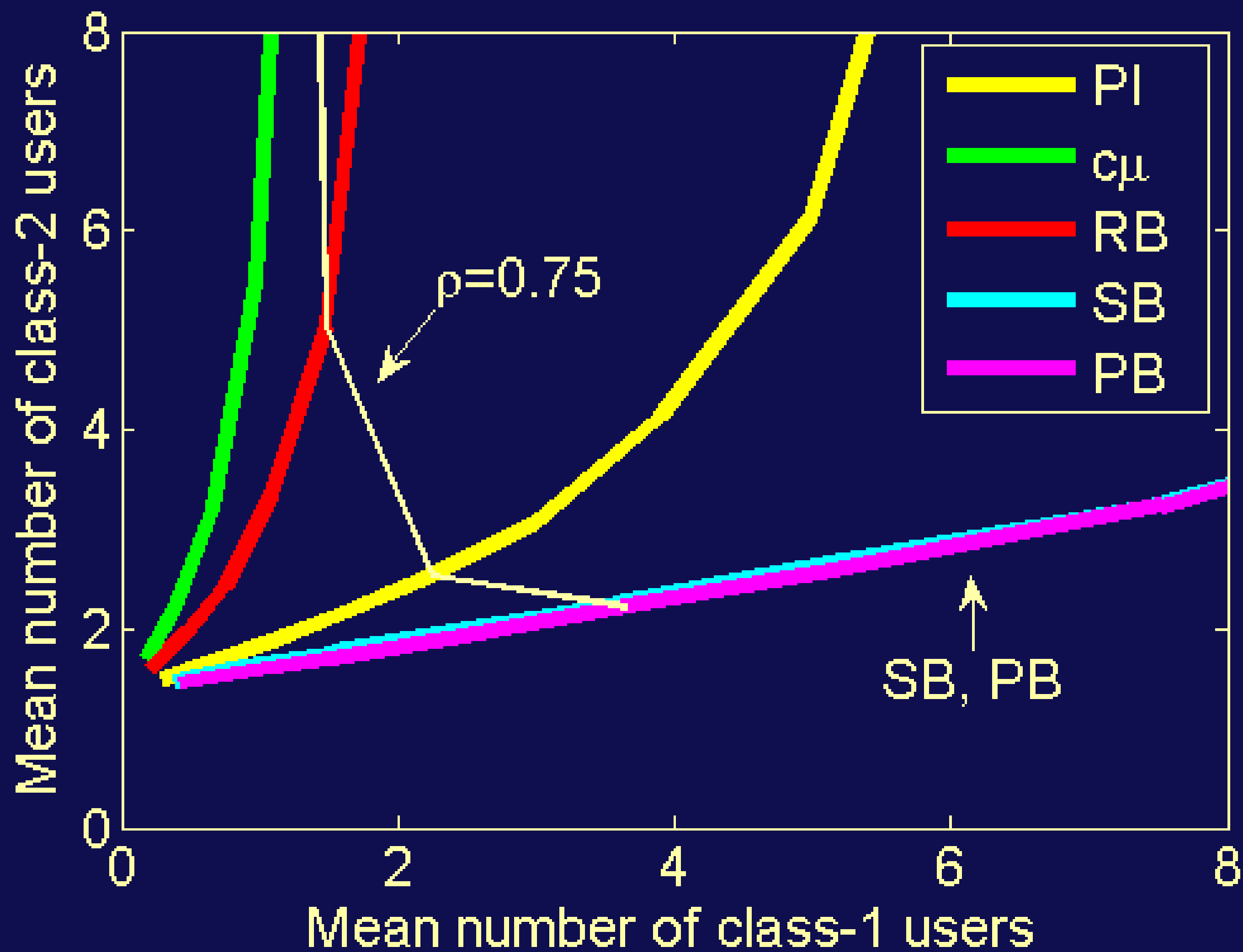
Numerical Simulations: Scenario 1

- Sample path of the number of users, $\varrho = 0.95$



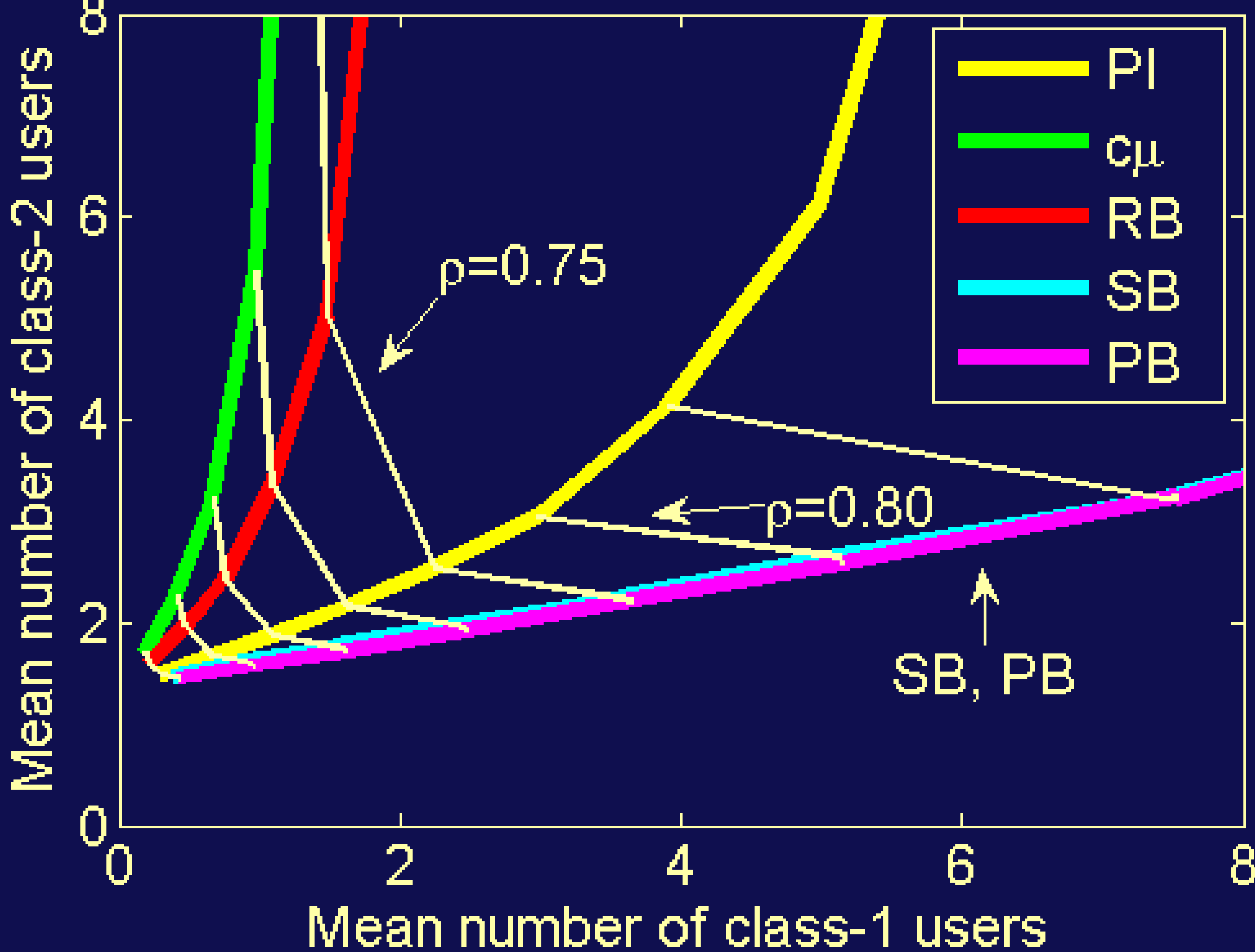
Numerical Simulations: Scenario 1

- Indifference curves for mean number of users



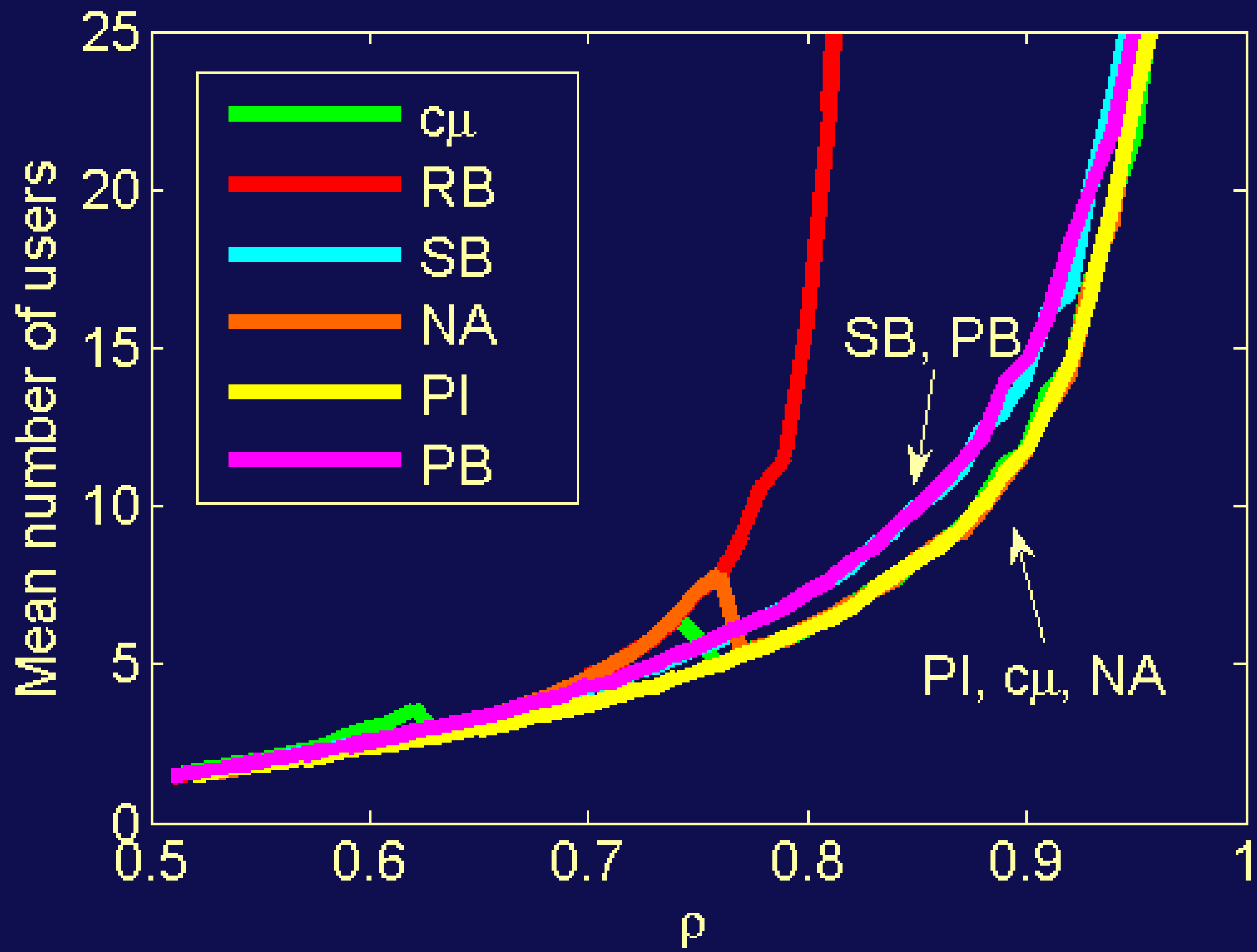
Numerical Simulations: Scenario 1

- Indifference curves for mean number of users



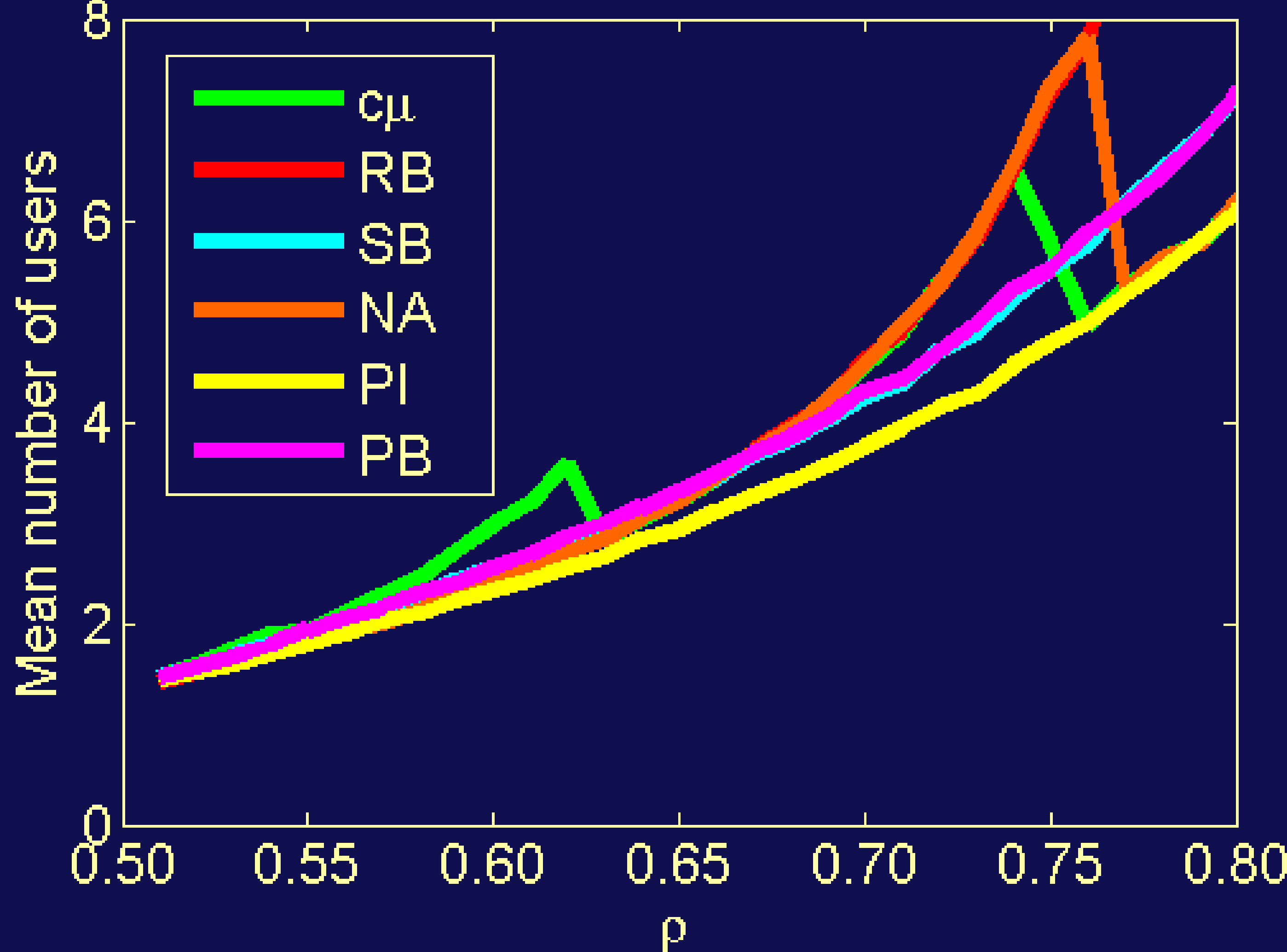
Numerical Simulations: Scenario 2

- Varied class-1 job length so that ϱ varies from 0.5 to 1



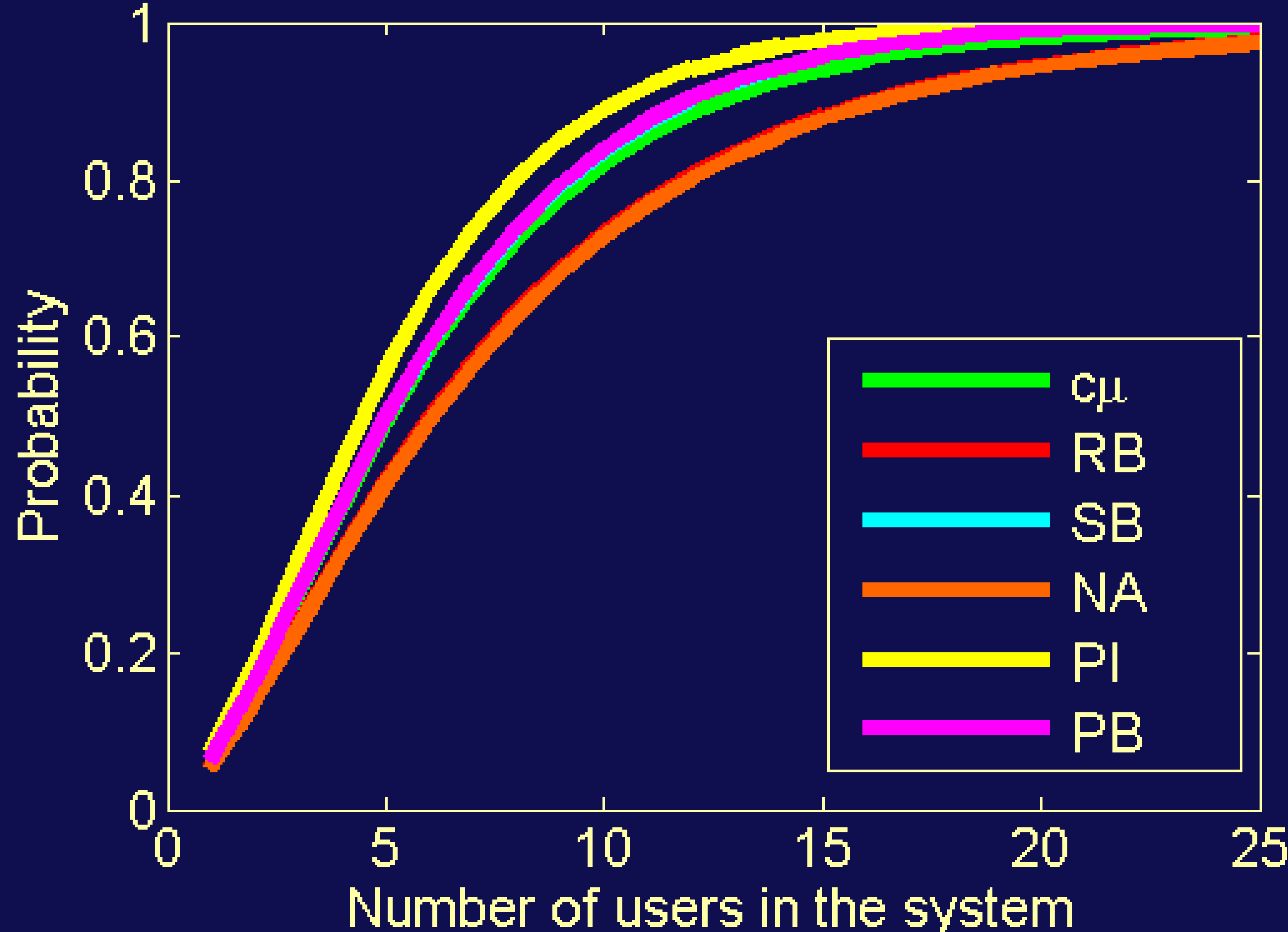
Numerical Simulations: Scenario 2

- Varied class-1 job length so that ϱ varies from 0.5 to 1



Numerical Simulations: Stoch. Dominance

- Typical picture of empirical CDFs



Simulations Summary

- PI consistently outperforms all the other rules
- Or its mean performance is equivalent to the best one
- Simulations strongly suggest stochastic dominance of PI over the other rules
- The stability region is the maximum for PI rule, while it is not for $c\mu$ and RB rules

Conclusion

- Framework to study opportunistic policies
  - ▷ RB (PF), PB roughly recovered under other rewards
- Tractable framework to obtain a new PI policy
  - ▷ asymptotically fluid-optimal (AEJV '10)
  - ▷ the only maximally stable policy in general (AL '10)
  - ▷ excellent performance in small-scale problems
- PI policy implies (roughly):
  - ▷ in low load: be channel-opportunistic
  - ▷ in high load: take into account job size ($c\mu$)

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

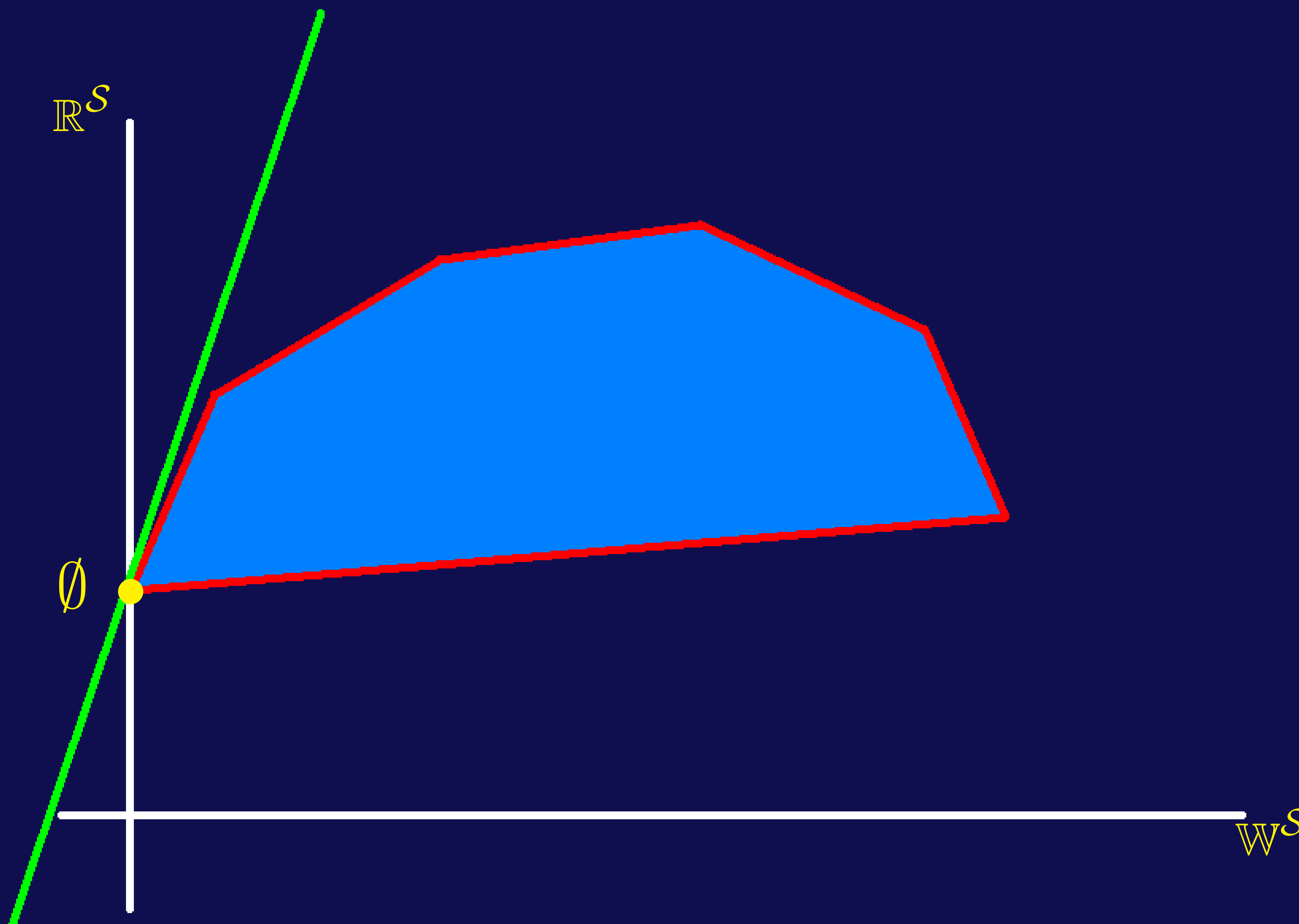
- For finite-state finite-action MDPs there exists an optimal policy that is deterministic, stationary, and independent of the initial state
  - ▷ we narrow our focus to those policies
  - ▷ represent them via serving sets $\mathcal{S} \subseteq \mathcal{N}$
  - ▷ policy $\mathcal{S}$ prescribes to serve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

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

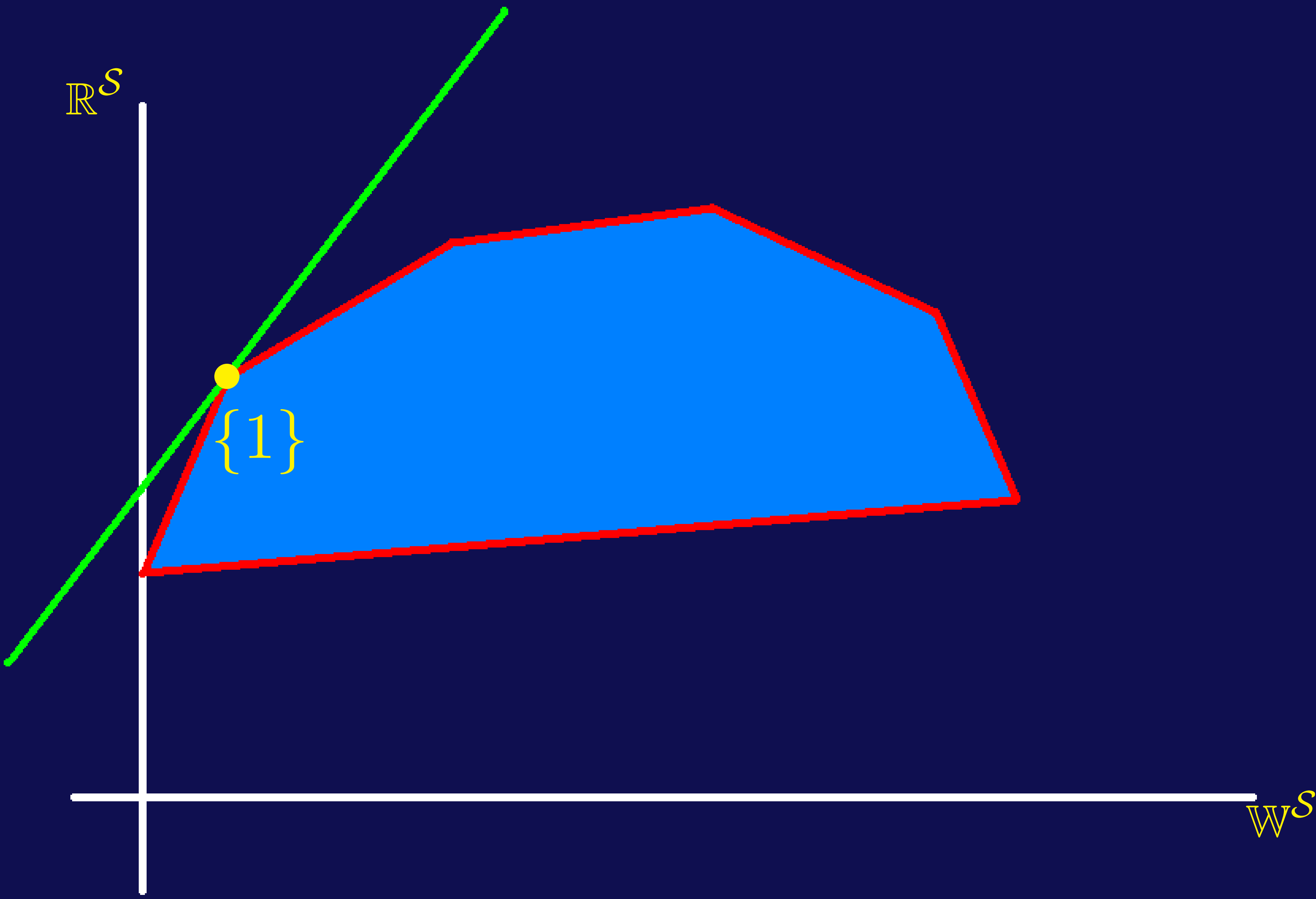
Geometric Interpretation

- $(\mathbb{W}_n^{\mathcal{S}}, \mathbb{R}_n^{\mathcal{S}})$ 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!

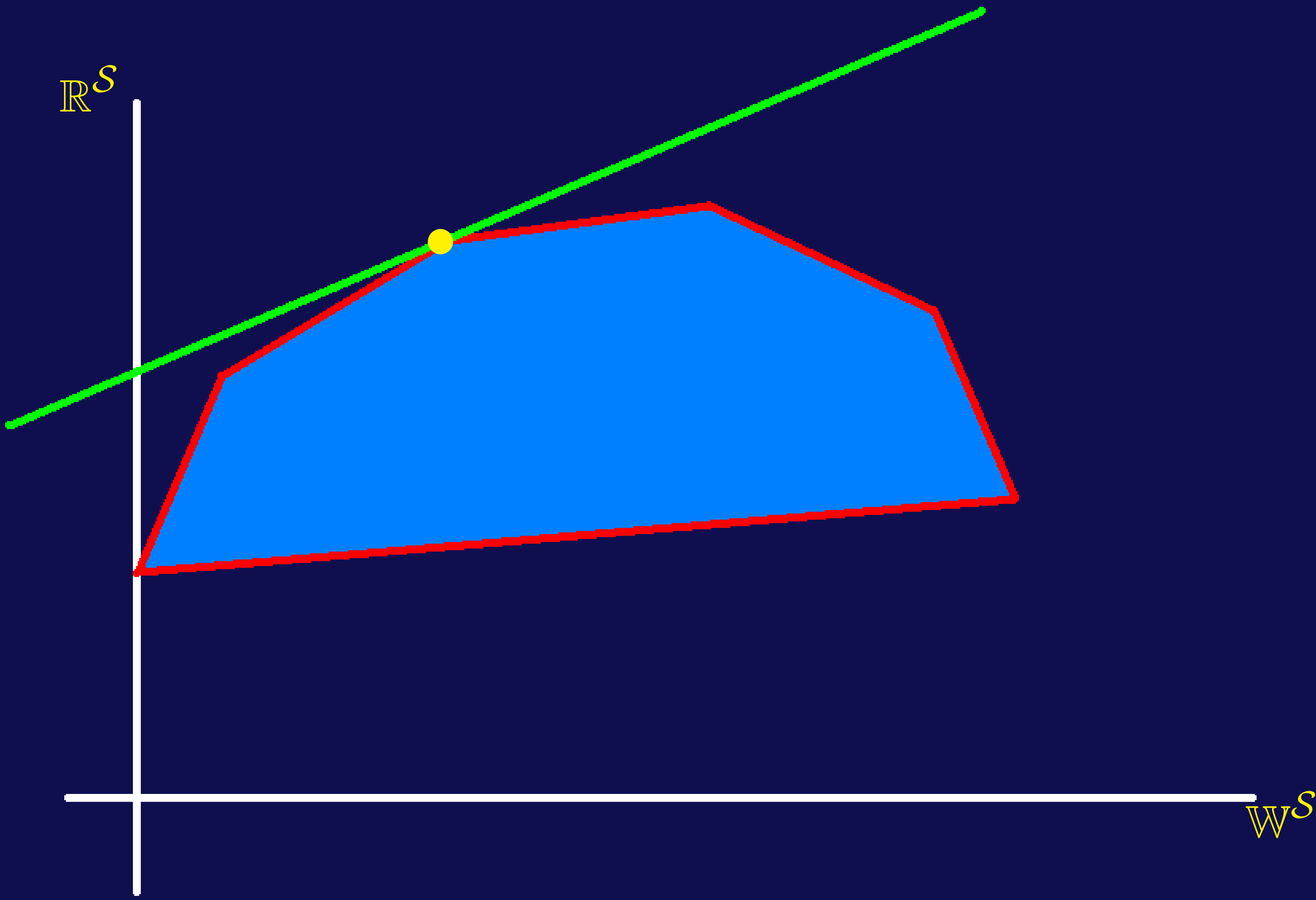
Performance Region



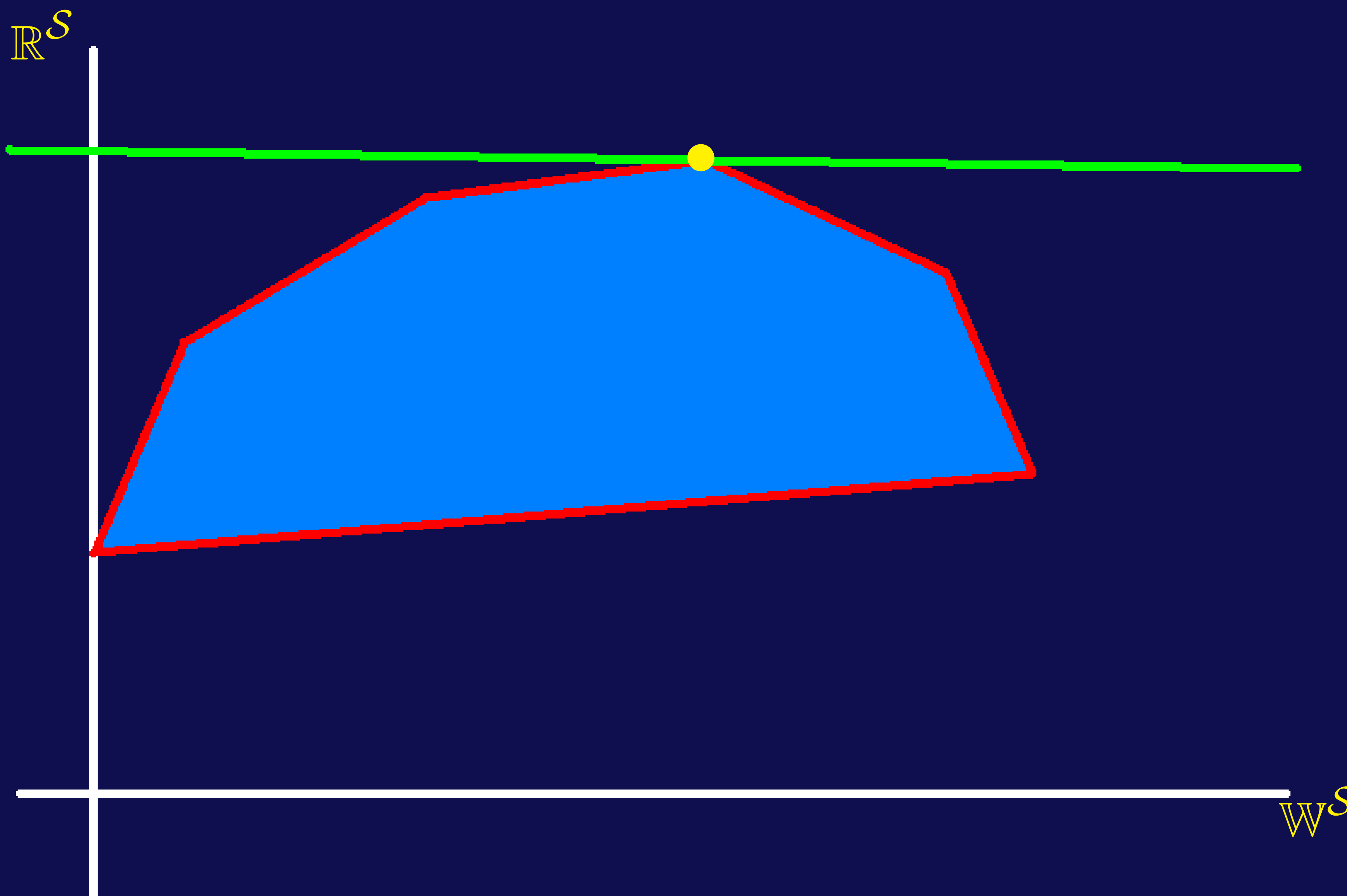
Performance Region



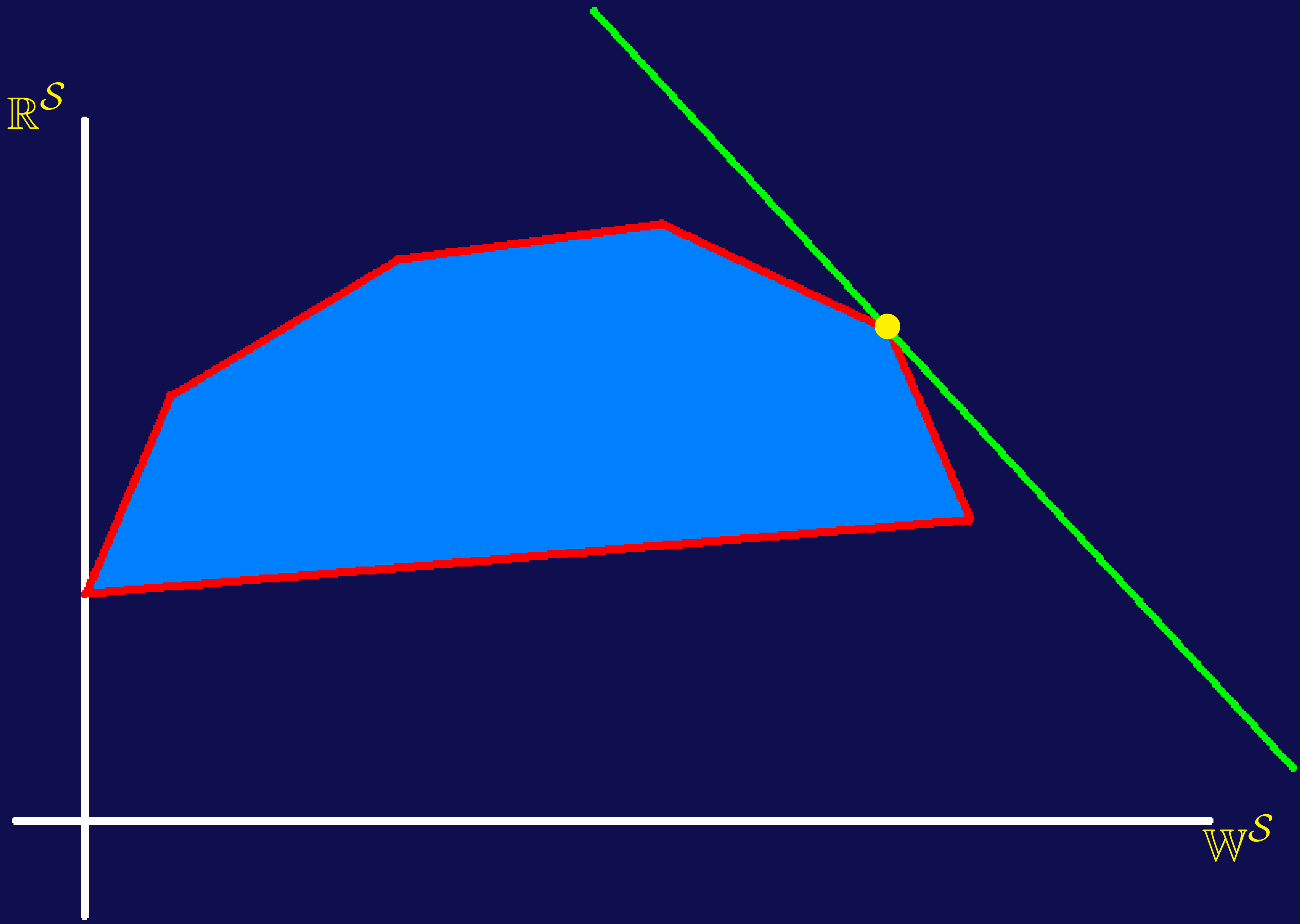
Performance Region



Performance Region



Performance Region



Performance Region