Nearly-Optimal Index Rules for Some Non-work-conserving Systems

Peter Jacko*
joint work with Urtzi Ayesta

INFORMS APS 2011, July 7

*BCAM — Basque Center for Applied Mathematics, Spain

Talk Outline

- Resource allocation MDP framework
- Decomposition and indexability
- Index rules for multi-class $-/Geo/M$ queues
  - ▷ $c\mu$-rule
  - ▷ $c(\mu - \theta)/\theta$-rule (for abandonments)
  - ▷ potential improvement rule (for time-varying rates)
  - ▷ a new rule (for time-varying and abandonments)
- Performance evaluation in systems with arrivals

Resource Allocation Problem (RAP)

- Stochastic and dynamic
- There is a number of independent competitors
- Constraint: resource capacity M at any time
- 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!

MDP Framework

- Markov Decision Processes / restless bandits
- Discrete time model $(t = 0, 1, 2, \ldots)$
- Job $k \in \mathcal{K}$ is defined by
  - ▷ states $\mathcal{N}_k$, actions $\mathcal{A} := \{0, 1\} = \{\text{'wait'}, \text{'serve'}\}$
  - ▷ expected one-period capacity consumption $\boldsymbol{W}_k^a$
  - ▷ 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

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 $\displaystyle\sum_{k\in\mathcal{K}} W_{k,X_k(t)}^{a_k(t)} = W,$ for all $t = 0, 1, 2, \ldots$

- Analogously under the time-average criterion
- This problem is PSPACE-hard (Papad. & Tsits. 1999)
  - ▷ intractable to solve exactly by Dynamic Programming
  - ▷ instead, we relax and decompose the problem

Relaxations and Decomposition

- 1. Whittle's (1988): Serve W jobs in expectation
  - ▷ infinite number of constraints is replaced by one
  - ▷ sort of perfect market assumption
- 2. Lagrangian: Pay cost ν for using the server
  - ▷ the constraint is moved into the objective
- Decomposes due to user independence into single-user parametric subproblems
  - ▷ solved by identifying the efficiency frontier
  - ▷ indexability $\approx$ threshold policies are optimal
  - ▷ math $+$ art $=$ characterize index values

Index Rules

- Assign an index value to each state of each user
- We are concerned with the following rule
  - ▷ at each time, be greedy:
    serve jobs with highest current index values
- In some problems it is optimal
  - ▷ $c\mu$-rule (Cox & Smith '61): job sequencing
  - ▷ Gittins index rule ('72): multi-armed bandit problem
  - ▷ Klimov index rule ('74): $M/G/1$ model w/ feedback
- Experiments and simulations suggest that it gives a nearly-optimal solution to RAP

Warm-up: Job Sequencing Problem

- Find a serving sequence minimizing the total cost of waiting of jobs $k \in \mathcal{K}$
  - ▷ c_k = cost of waiting for job k
  - ▷ μ_k = completion probability for job k
- $\mathcal{N}_k := \{\text{‘completed’}, \text{‘waiting’}\}$, $\mathcal{A} := \{\text{‘serve’}, \text{‘wait’}\}$
- Expected one-period capacity consumption

$$W_{k,\text{‘completed’}}^{\text{‘serve’}} := 1, \qquad W_{k,\text{‘waiting’}}^{\text{‘serve’}} := 1,$$
$$W_{k,\text{‘completed’}}^{\text{‘wait’}} := 0, \qquad W_{k,\text{‘waiting’}}^{\text{‘wait’}} := 0;$$

Warm-up: Job Sequencing Problem

- Expected one-period reward

$$R_{k,\text{'completed'}}^{\text{'serve'}} := 0, \qquad R_{k,\text{'waiting'}}^{\text{'serve'}} := -c_k(1-\mu_k),$$
$$R_{k,\text{'completed'}}^{\text{'wait'}} := 0, \qquad R_{k,\text{'waiting'}}^{\text{'wait'}} := -c_k;$$

- One-period transition probability matrices

$$\boldsymbol{P}_k^{\text{'serve'}} := \begin{array}{r} \\ \text{'completed'} \\ \text{'waiting'} \end{array} \begin{array}{c} \begin{array}{cc} \text{'completed'} & \text{'waiting'} \end{array} \\ \begin{pmatrix} 1 & 0 \\ \mu_k & 1-\mu_k \end{pmatrix} \end{array}$$

$$\boldsymbol{P}_k^{\text{'wait'}} := \begin{array}{r} \\ \text{'completed'} \\ \text{'waiting'} \end{array} \begin{array}{c} \begin{array}{cc} \text{'completed'} & \text{'waiting'} \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

JSP with Abandonments

- Find a serving sequence minimizing the total cost of waiting and abandonment penalties of jobs $k \in \mathcal{K}$
  - ▷ c_k = cost of waiting for job k
  - ▷ μ_k = completion probability for job k
  - ▷ d_k = abandonment penalty for job k
  - ▷ θ_k = abandonment probability for job k
- $\mathcal{N}_k := \{$'completed or abandoned', 'waiting'$\}$
- $\mathcal{A} := \{$'wait', 'serve'$\}$
- Expected one-period capacity consumption ...

JSP with Abandonments

- Expected one-period reward

$$R_{k,\text{'completed or abandoned'}}^{\text{'serve'}} := 0, \quad R_{k,\text{'waiting'}}^{\text{'serve'}} := -c_k(1-\mu_k),$$
$$R_{k,\text{'completed or abandoned'}}^{\text{'wait'}} := 0, \quad R_{k,\text{'waiting'}}^{\text{'wait'}} := -c_k(1-\theta_k) - d_k\theta_k;$$

- One-period transition probability matrices

$$\boldsymbol{P}_k^{\text{'serve'}} := \begin{array}{r} \\ \text{'compl. or ab.'} \\ \text{'waiting'} \end{array} \begin{array}{c} \begin{array}{cc} \text{'compl. or ab.'} & \text{'waiting'} \end{array} \\ \begin{pmatrix} 1 & 0 \\ \mu_k & 1-\mu_k \end{pmatrix} \end{array}$$

$$\boldsymbol{P}_k^{\text{'wait'}} := \begin{array}{r} \\ \text{'compl. or ab.'} \\ \text{'waiting'} \end{array} \begin{array}{c} \begin{array}{cc} \text{'compl. or ab.'} & \text{'waiting'} \end{array} \\ \begin{pmatrix} 1 & 0 \\ \theta_k & 1-\theta_k \end{pmatrix} \end{array}$$

JSP with Abandonments

- Under discounted criterion:

$$\nu^{\text{AJN}}_{k,\text{‘waiting’}} = \frac{c_k(\mu_k - \theta_k) + d_k\theta_k(1 - \beta + \beta\mu_k)}{1 - \beta + \beta\theta_k}$$

$$\nu^{\text{AJN}}_{k,\text{‘completed or abandoned’}} = 0$$

- Under time-average criterion:

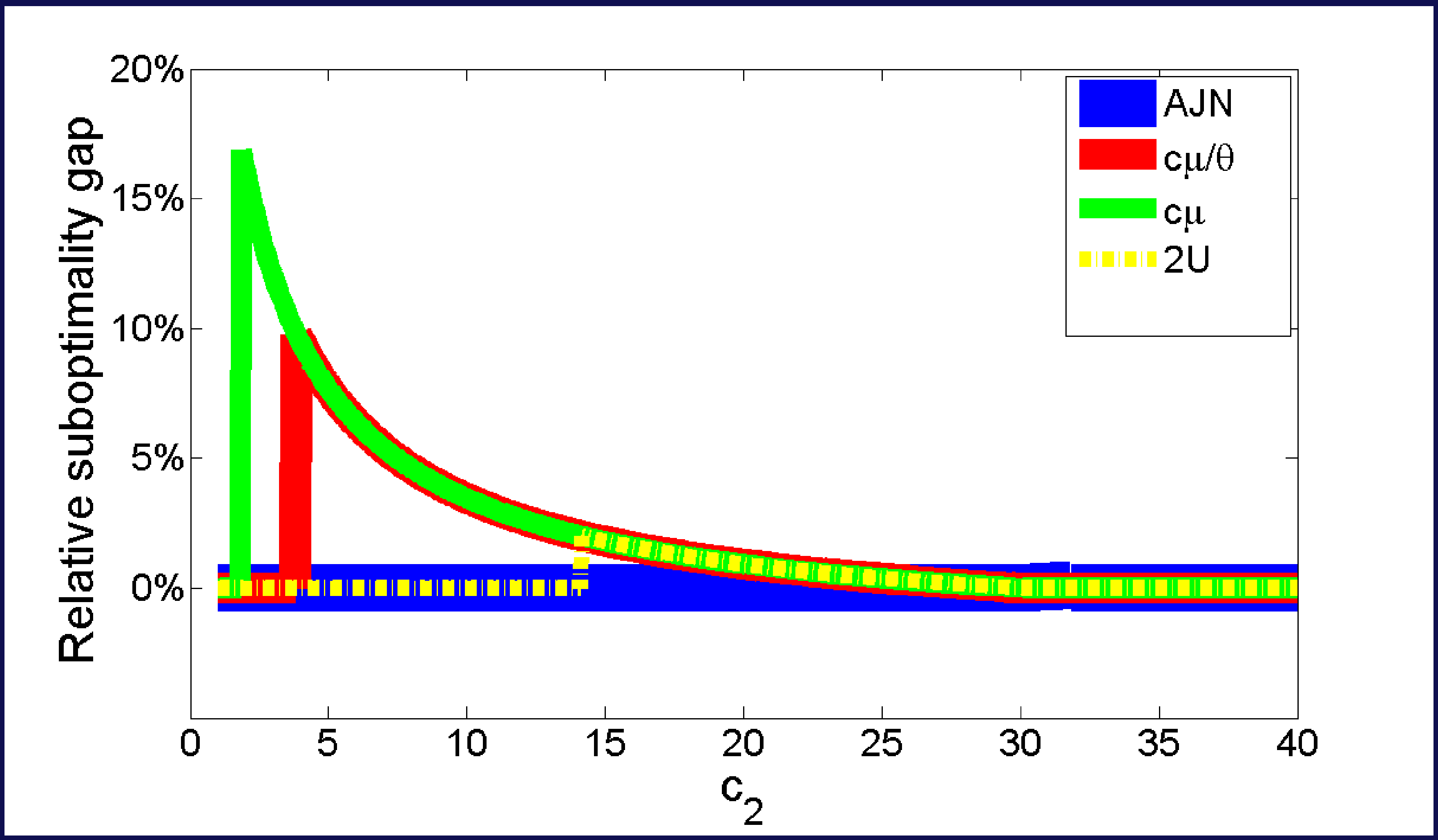
$$\nu^{\text{AJN}}_{k,\text{‘waiting’}} = \frac{c_k(\mu_k - \theta_k) + d_k\mu_k\theta_k}{\theta_k}$$

$$\nu^{\text{AJN}}_{k,\text{‘completed or abandoned’}} = 0$$

- Ayesta, J. & Novak (2011), IEEE INFOCOM

JSP with Abandonments

- Two classes: $\mu_1 = 0.4$, $\mu_2 = 0.22$, $\theta_1 = 0.1$, $\theta_2 = 0.2$ and $c_1 = d_1 = d_2 = \lambda_1 = \lambda_2 = 1$



JSP with iid Time-Varying Service Rates

- Job/user $k \in \mathcal{K}$ is defined by
  - ▷ $c_k =$ cost of waiting for job k
  - ▷ $q_{k,n} =$ probability to move to condition n
  - ▷ $\mu_{k,n} =$ completion probability for job k under condition n (ordered: $\mu_{k,n} \leq \mu_{k,n+1}$)
- Find a serving sequence minimizing the total cost of waiting of jobs $k \in \mathcal{K}$
- $\mathcal{N}_k := \{0, 1, 2, \ldots, N_k\}$, $\mathcal{A} := \{$'wait', 'serve'$\}$
- $0 =$ 'completed' ; $n =$ 'waiting' and condition is n

JSP with iid Time-Varying Service Rates

- Expected one-period reward

$$R_{k,0}^{\text{`serve'}} := 0, \qquad R_{k,n}^{\text{`serve'}} := -c_k(1-\mu_{k,n}),$$
$$R_{k,0}^{\text{`wait'}} := 0, \qquad R_{k,n}^{\text{`wait'}} := -c_k;$$

- One-period transition probability matrices

$$\boldsymbol{P}_k^{\text{`serve'}} := \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ N_k \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & \dots & N_k \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ \mu_{k,1} & \widetilde{\mu}_{k,1} q_{k,1} & \dots & \widetilde{\mu}_{k,1} q_{k,N_k} \\ \mu_{k,2} & \widetilde{\mu}_{k,2} q_{k,1} & \dots & \widetilde{\mu}_{k,2} q_{k,N_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{k,N_k} & \widetilde{\mu}_{k,N_k} q_{k,1} & \dots & \widetilde{\mu}_{k,N_k} q_{k,N_k} \end{pmatrix} \end{array}$$

JSP with iid Time-Varying Service Rates

- Potential improvement (opportunistic) index
- Under discounted criterion:

$$\nu_{k,n}^{\mathsf{PI}} = \frac{c_k \mu_{k,n}}{(1-\beta) + \beta \sum_{m>n} q_{k,m}(\mu_{k,m} - \mu_{k,n})}$$

- Under time-average criterion:

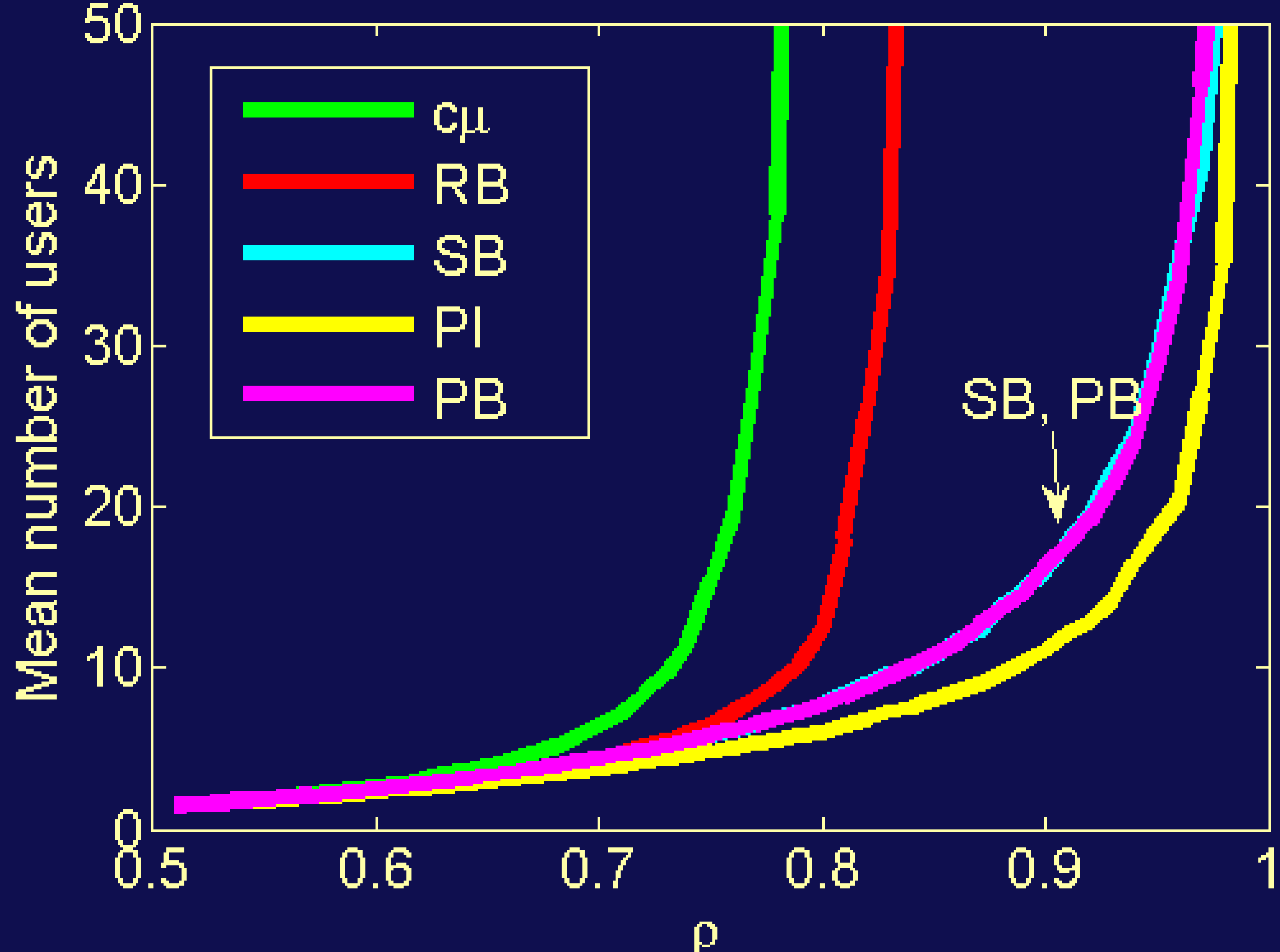
$$\nu_{k,n}^{\mathsf{PI}} = \frac{c_k \mu_{k,n}}{\sum_{m>n} q_{k,m}(\mu_{k,m} - \mu_{k,n})} \text{ for } n \neq N_k, \qquad \nu_{k,N_k}^{\mathsf{PI}} = \infty$$

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

- Ayesta, Erausquin & J. (2010), IFIP Performance

JSP with iid Time-Varying Service Rates

- Varied λ_1 so that ϱ varies from 0.5 to 1



with Markov Time-Varying Service Rates

- Job/user $k \in \mathcal{K}$ is defined by
  - c_k = cost of waiting for job k
  - $q_{k,n,m}$ = probability to move from condition n to m
  - $\mu_{k,n}$ = completion probability for job k under condition n (ordered: $\mu_{k,n} \leq \mu_{k,n+1}$)
- Find a serving sequence minimizing the total cost of waiting of jobs $k \in \mathcal{K}$
- Gilbert-Elliot conditions: bad (B), good (G)
- $\mathcal{N}_k := \{0, B, G\}$, $\mathcal{A} := \{$'wait', 'serve'$\}$
- $0 =$ 'completed' ; $n =$ 'waiting' and condition is n

with Markov Time-Varying Service Rates

- Expected one-period reward

$$R_{k,0}^{\text{`serve'}} := 0, \qquad R_{k,n}^{\text{`serve'}} := -c_k(1-\mu_{k,n}),$$
$$R_{k,0}^{\text{`wait'}} := 0, \qquad R_{k,n}^{\text{`wait'}} := -c_k;$$

- One-period transition probability matrices

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

with Markov Time-Varying Service Rates

- Generalized Potential improvement index
- Under discounted criterion:

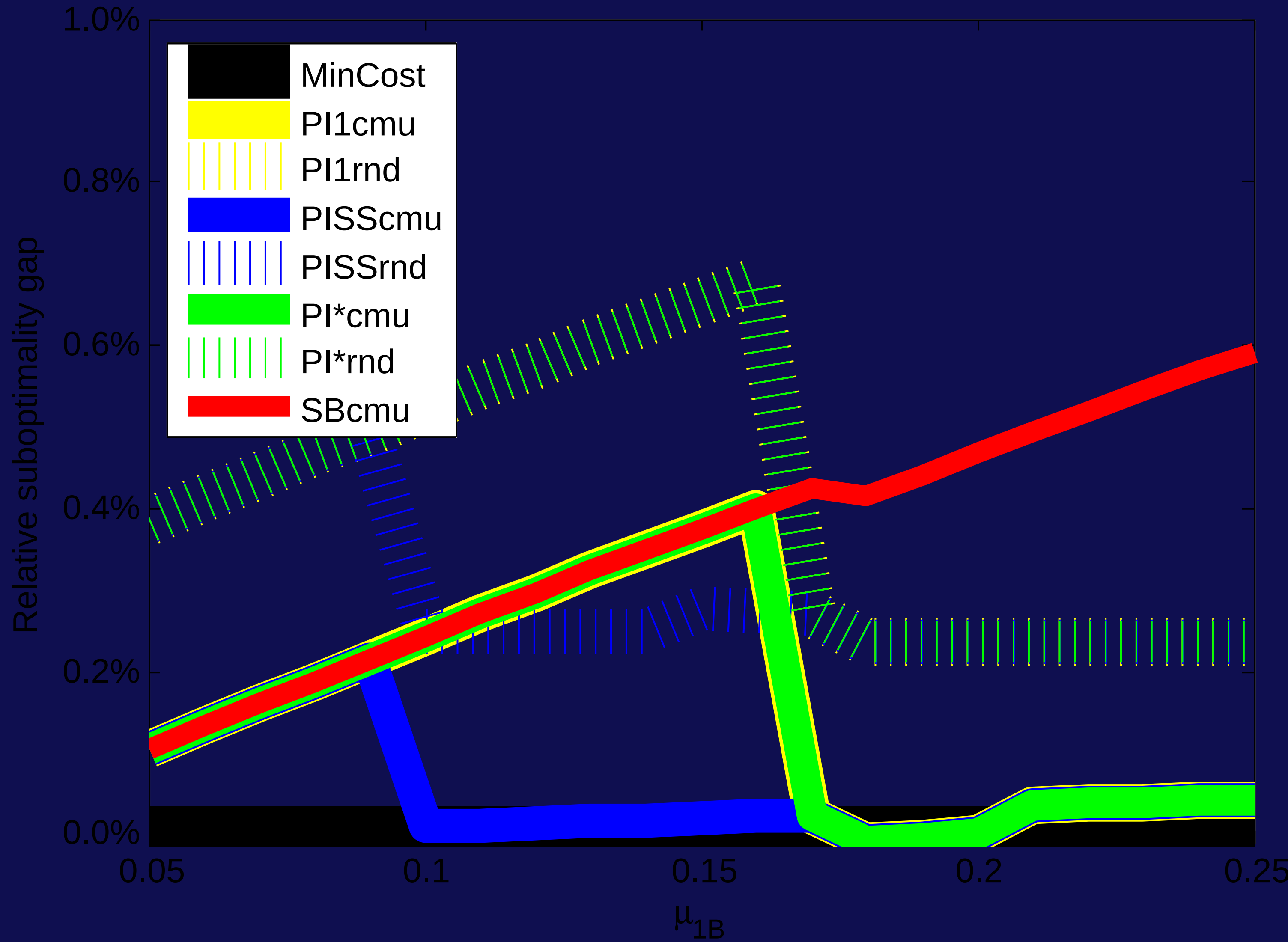
$$\nu^*_{k,G} = \frac{c_k \mu_{k,G}}{(1-\beta)}, \qquad \nu^*_{k,B} = \frac{c_k \mu_{k,B}}{(1-\beta) + \beta q^*_{k,B,G}(\mu_{k,G} - \mu_{k,B})}$$

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

- J. (2011), IFIP Performance

with Markov Time-Varying Service Rates

- Varied $\mu_{1,B}$, while $\mu_{1,G} = 1$



with iid Time-Varying and Abandonments

- Job/user $k \in \mathcal{K}$ is defined by
  - ▷ c_k = cost of waiting for job k
  - ▷ $q_{k,n}$ = probability to move to condition n
  - ▷ $\mu_{k,n}$ = completion prob. for job k under condition n
  - ▷ d_k = abandonment penalty for job k
  - ▷ θ_k = abandonment probability for job k
- Find a serving sequence minimizing the total cost of waiting and abandonment penalties of jobs $k \in \mathcal{K}$
- $\mathcal{N}_k := \{0, 1, 2, \ldots, N_k\}$, $\mathcal{A} :=$ {'wait', 'serve'}
  - ▷ $0 =$ 'completed or abandoned';
    $n =$ 'waiting' and condition is n

with iid Time-Varying and Abandonments

- Expected one-period reward

$$R_{k,0}^{\text{'serve'}} := 0, \qquad R_{k,n}^{\text{'serve'}} := -c_k(1-\mu_{k,n}),$$
$$R_{k,0}^{\text{'wait'}} := 0, \qquad R_{k,n}^{\text{'wait'}} := -c_k(1-\theta_k) - d_k\theta_k;$$

- One-period transition probability matrices

$$\boldsymbol{P}_k^{\text{'wait'}} := \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ N_k \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & \dots & N_k \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ \theta_{k,1} & \widetilde{\theta}_{k,1} q_{k,1} & \dots & \widetilde{\theta}_{k,1} q_{k,N_k} \\ \theta_{k,2} & \widetilde{\theta}_{k,2} q_{k,1} & \dots & \widetilde{\theta}_{k,2} q_{k,N_k} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{k,N_k} & \widetilde{\theta}_{k,N_k} q_{k,1} & \dots & \widetilde{\theta}_{k,N_k} q_{k,N_k} \end{pmatrix} \end{array}$$

with iid Time-Varying and Abandonments

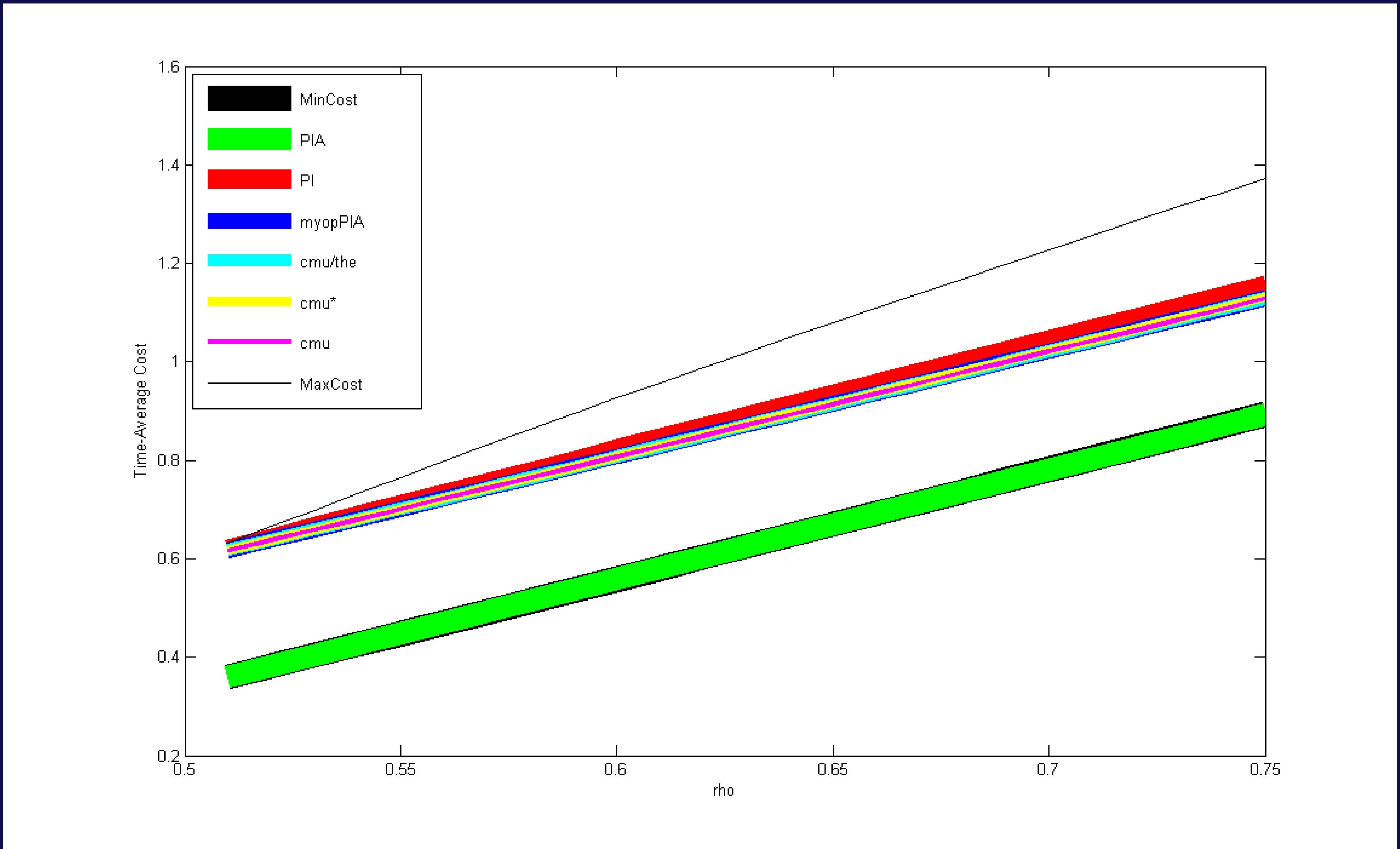
- A new (opportunistic) index
- Under time-average criterion:

$$\nu_{k,n}^{\mathsf{PIA}} = \frac{c_k(\mu_{k,n} - \theta_k) + d_k\theta_k \left(\mu_{k,n} + \sum_{m>n} q_{k,m}(\mu_{k,m} - \mu_{k,n}) \right)}{\theta_k + \sum_{m>n} q_{k,m}(\mu_{k,m} - \mu_{k,n})}$$

- Recovers PI and AJN indices

with iid Time-Varying and Abandonments

- Varied λ_1 so that ϱ varies from 0.5 to 0.75



Conclusion

- Rich framework to study scheduling problems
  - ▷ obtain elegant index rules
  - ▷ index policies optimal for relaxations
  - ▷ suggests structure of (asymptotically) optimal policies
- Weakness
  - ▷ no stability/optimality results
- Open problems
  - ▷ non-geometric job sizes
  - ▷ optimal solution (structure)
  - ▷ correlation among users

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

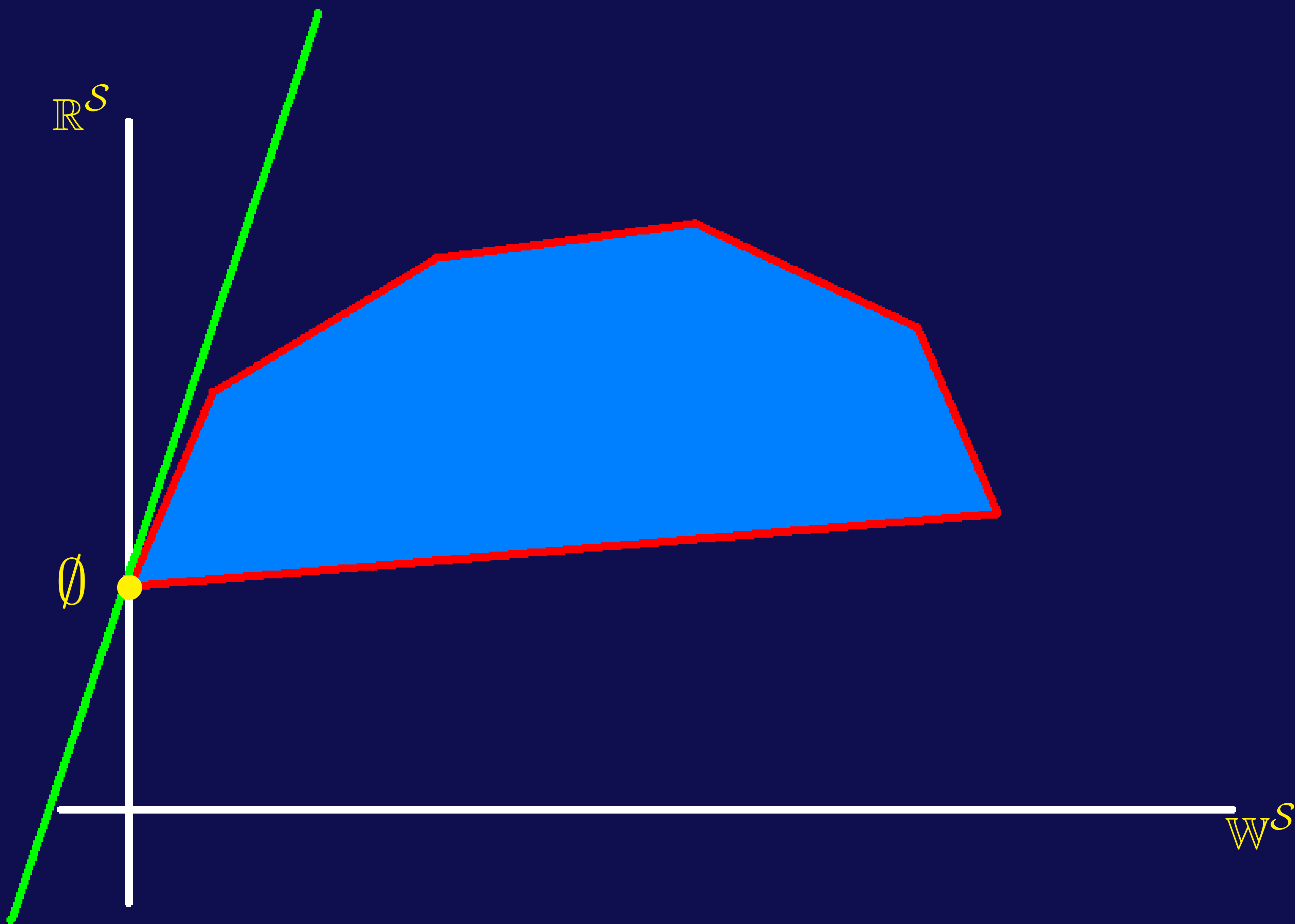
- For finite-state finite-action MDPs there exists an optimal policy that is deterministic, stationary, and independent of the initial state
  - ▷ we narrow our focus to those policies
  - ▷ represent them via serving sets $\mathcal{S} \subseteq \mathcal{N}$
  - ▷ policy $\mathcal{S}$ prescribes to serve 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

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

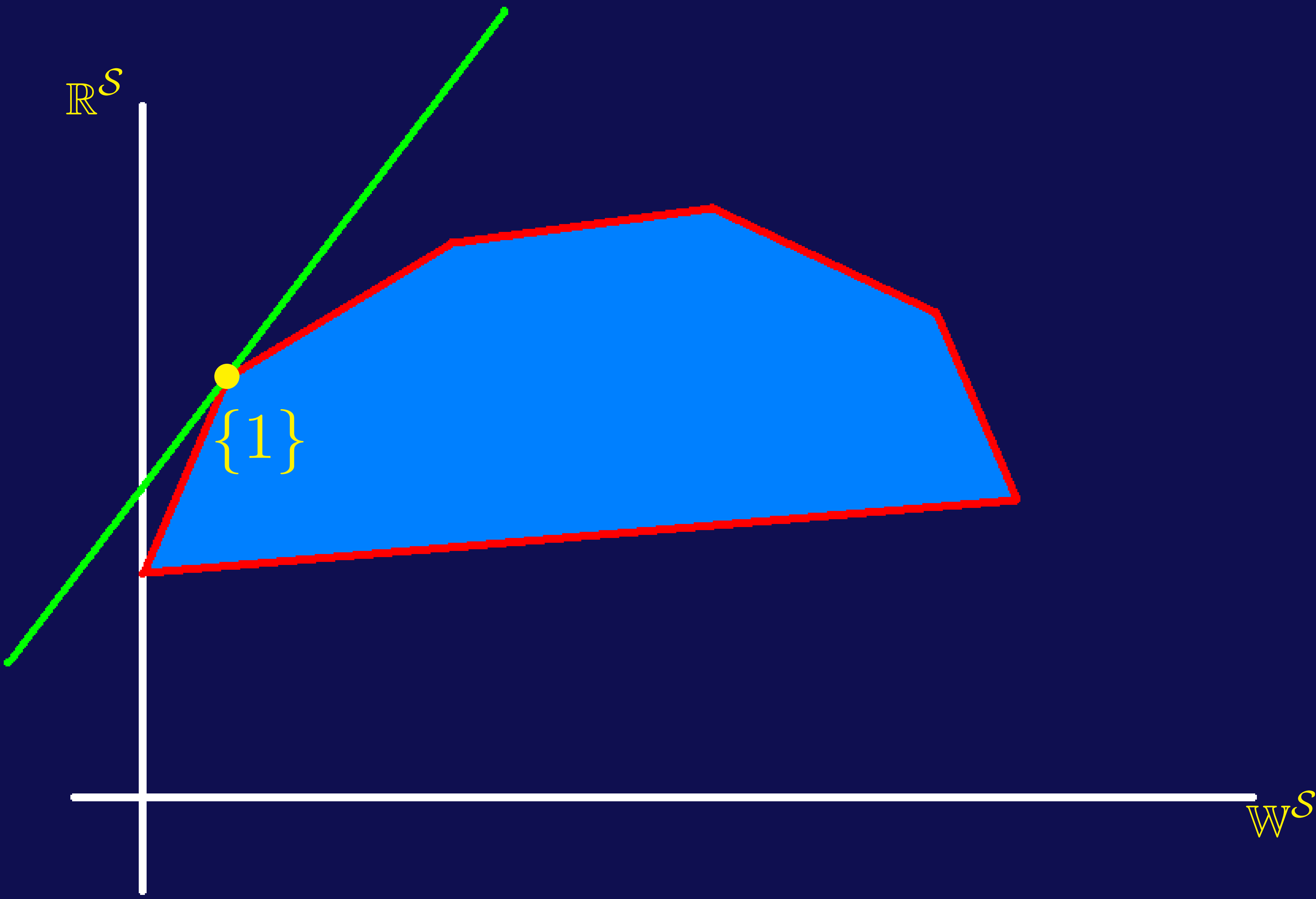
Geometric Interpretation

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

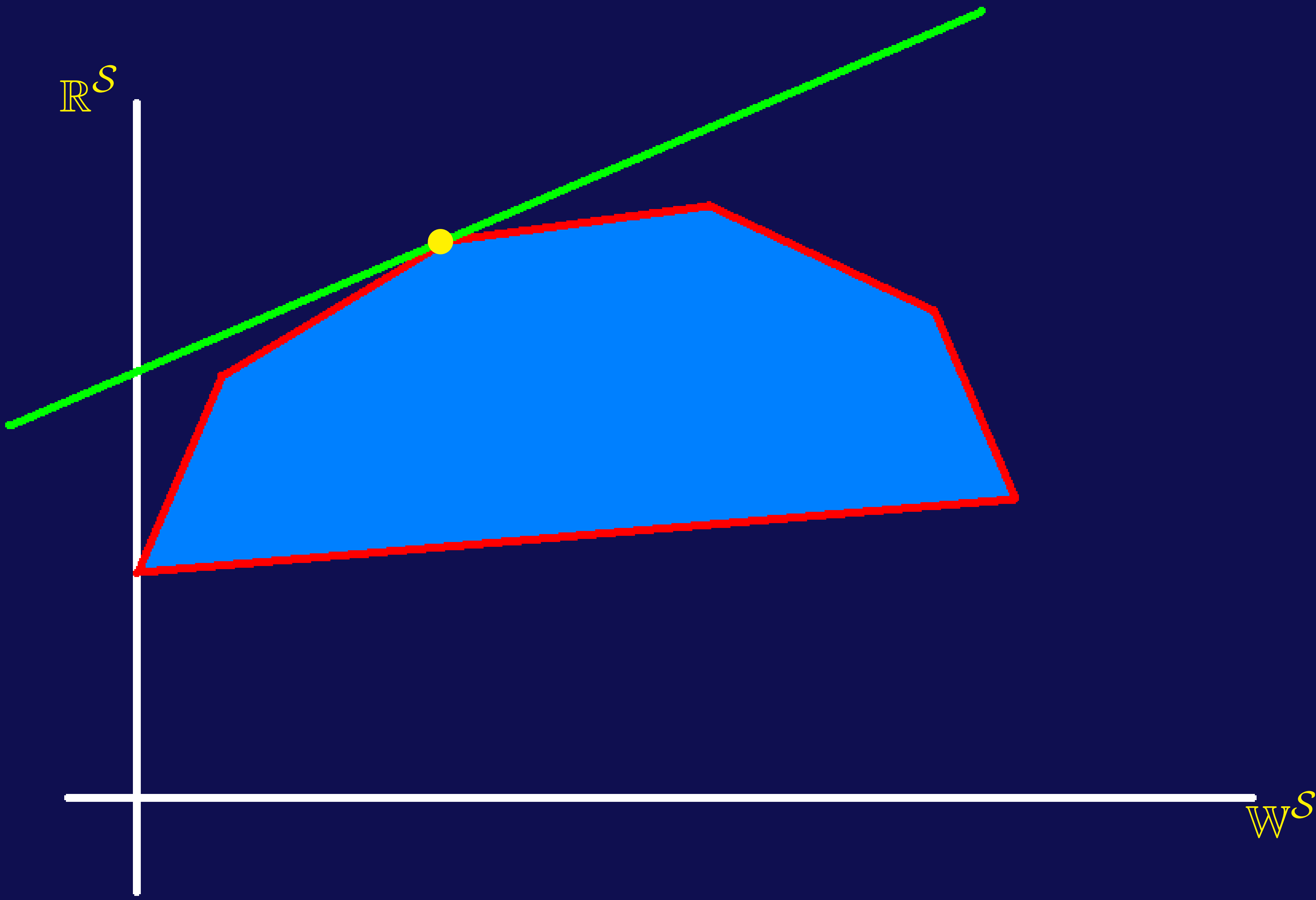
Performance Region



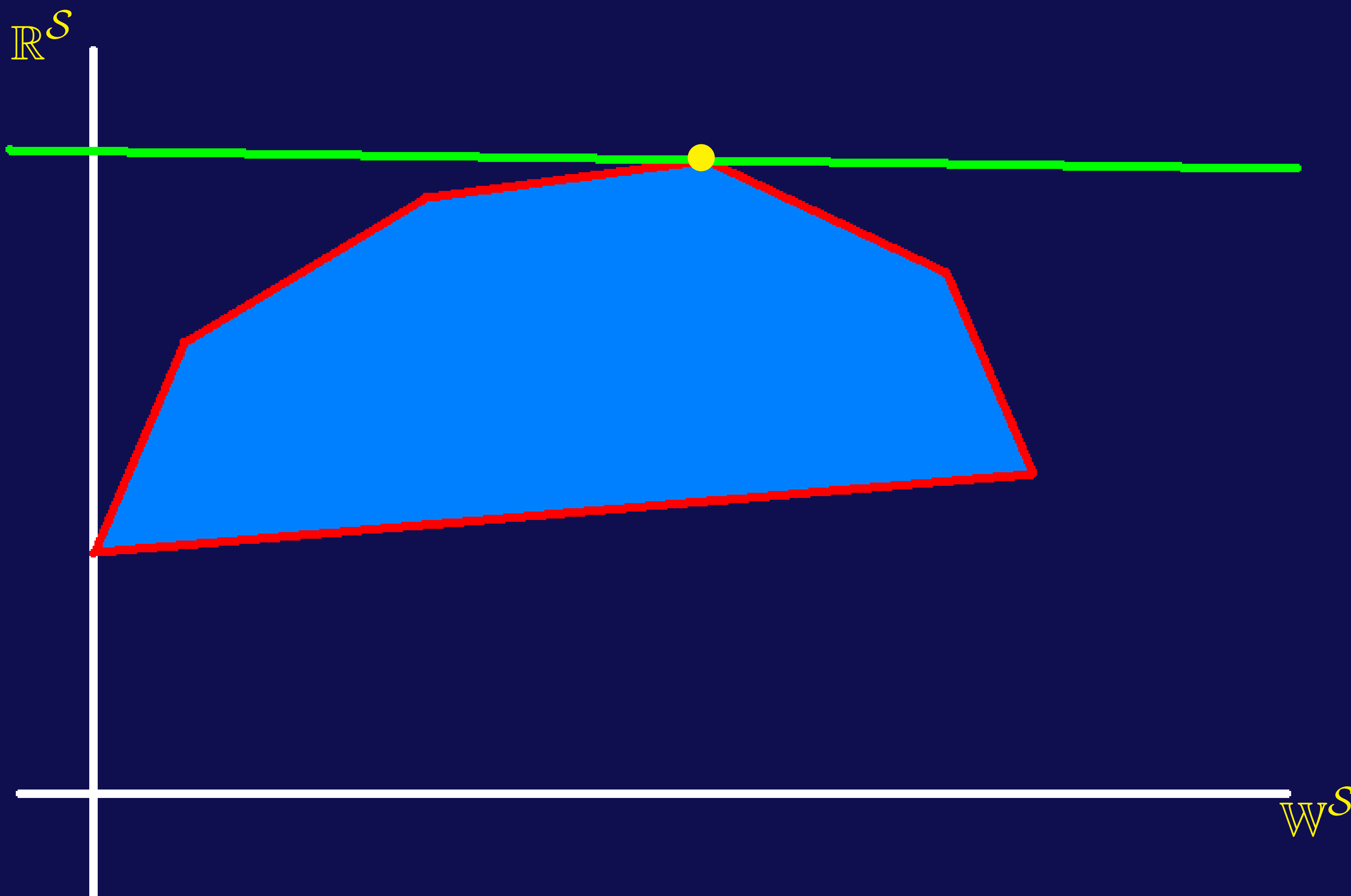
Performance Region



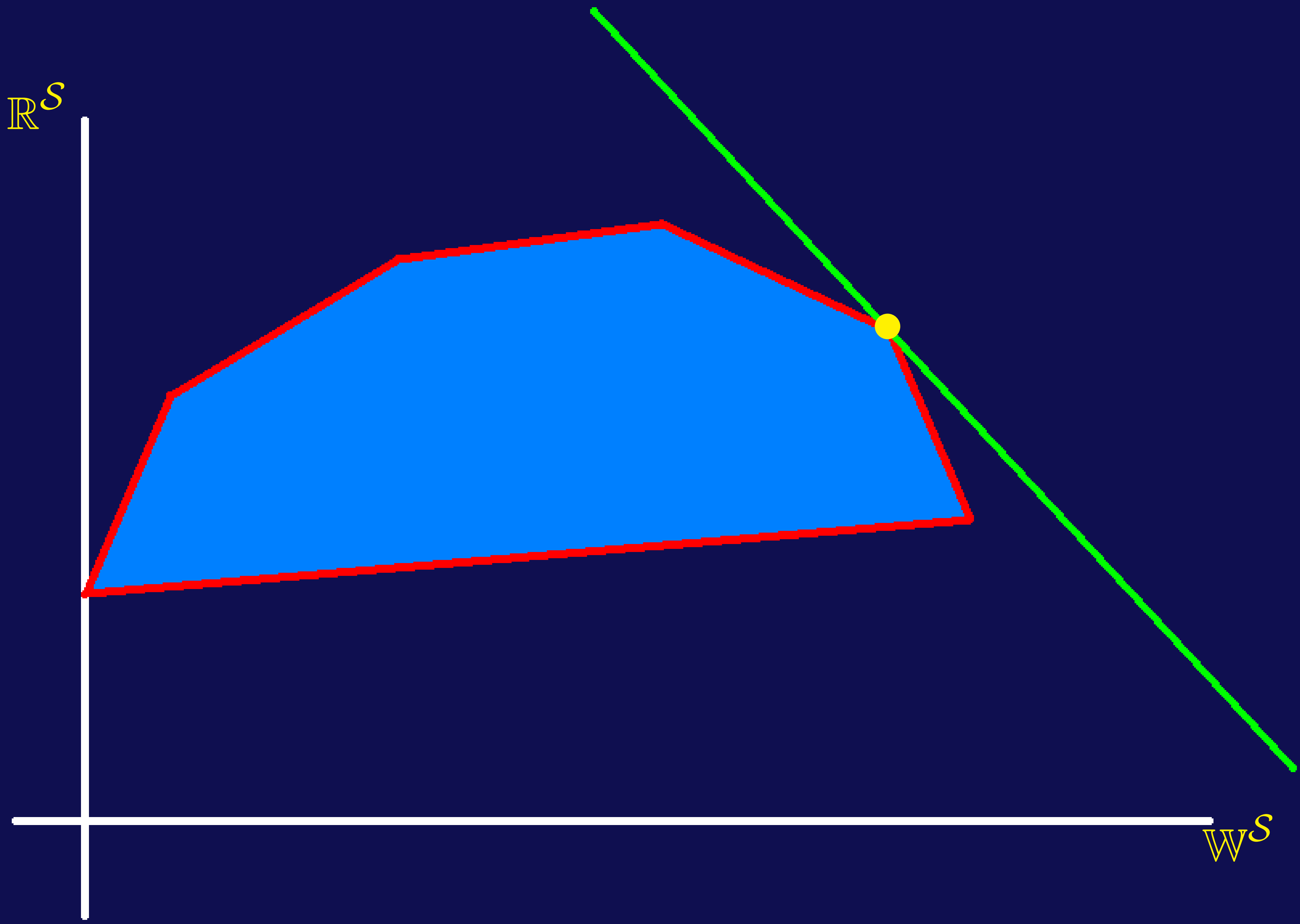
Performance Region



Performance Region



Performance Region



Performance Region