# A Nearly-Optimal Index Rule for Scheduling of Users with Abandonment

Vladimir Novak[*,‡]
joint work with Urtzi Ayesta[*,†] and Peter Jacko[*]

[*]BCAM - Basque Center for Applied Mathematics, Bilbao, Spain

[†]IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

[‡]Faculty of Mathematics, Physics and Informatics.
Comenius University in Bratislava, Slovakia

IEEE INFOCOM 2011, Shanghai, China

# Abandonments is a ubiquitous phenomenon

- Abandonment happens in multitude of systems
  - Customers waiting too long in queue
  - User's with too slow Internet connection
- Very negative impact on system performance
  - Users consider that the system is poorly managed
  - Waste of resources
- Scheduling of impatient users is not completely understood, because of its complexity

# Motivation

- *Main focus*: Design a scheduling rule to minimize the total discounted or time-average cost

- *Methodology:* Recent developments on restless bandits

# Talk Outline

1. Problem Description
2. MDP Formulation
3. Analytical solution - special cases
4. Index policies - general case
5. Computational Experiments

# Problem Description

- Fixed number of jobs waiting for service
- Server
  - serves one job at a time
  - preemptive
  - regularly decides to which user (if any) it should be allocated
- Job $k$:
  - completed with probability $\mu_k > 0$
  - abandoned with probability $\theta_k \geq 0$
  - holding cost $c_k > 0$
  - abandonment penalty $d_k > 0$, if user abandons the system without having the job completed
- User in service cannot abandon
- It is allowed to idle the server even if there are users waiting

# MDP Formulation

- The time slotted into epochs $t \in \mathcal{T} := \{0, 1, 2, \dots\}$
- User $k$ is defined by
  - action space $\{0, 1\} = \{\text{"do not serve"}, \text{"serve"}\}$
  - state space $\{0, 1\} = \{\text{"departed"}, \text{"waiting"}\}$
  - expected one-period server utilization

$$W_{k,n}^{1} := 1, \qquad W_{k,n}^{0} := 0;$$

  - expected one-period reward

$$R_{k,0}^{1} := 0, \qquad R_{k,1}^{1} := -c_k \cdot (1 - \mu_k) + 0 \cdot \mu_k,$$
$$R_{k,0}^{0} := 0, \qquad R_{k,1}^{0} := -c_k \cdot (1 - \theta_k) - d_k \cdot \theta_k;$$

# MDP Formulation

- 
  - one-period transition probability matrix

$$
\boldsymbol{P}_k^1 := \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 1 & 0 \\ \mu_k & 1-\mu_k \end{pmatrix} \end{array}, \quad \boldsymbol{P}_k^0 := \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 1 & 0 \\ \theta_k & 1-\theta_k \end{pmatrix} \end{array}
$$

  - State process $X_k(t)$ and action process $a_k(t)$
- Infinite- horizon $\beta$-average quantity:

$$
\mathbb{B}_0^\pi \left[ Q_{X(\cdot)}^{a(\cdot)}, \beta \right]
$$

  - $\beta = 1$ expected time-average quantity
  - $0 < \beta < 1$ expected total $\beta$-discounted quantity
  - $\beta = 0$ myopic quantity

# MDP Formulation - Optimization Problem

- Formulation under the $\beta$-average criterion

$$\max_{\boldsymbol{\pi}\in\Pi_{X,\boldsymbol{a}}} \mathbb{B}_0^{\boldsymbol{\pi}}\left[\sum_{k\in\mathcal{K}} R_{k,X_k(\cdot)}^{a_k(\cdot)}\right] \qquad \text{(P)}$$

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

  - intractable to solve exactly by Dynamic Programming
  - a restless bandit problem $\rightarrow$ PSPACE-hard (Papadimitriou and Tsitsiklis, 1999)

# Special Cases - 1U index

- Case of single user competing with idling the server
- 1U index:

$$\nu_k^{1\mathrm{U}} := c_k(\mu_k - \theta_k) + d_k\theta_k(1 - \beta + \beta\mu_k).$$

- *Proposition:*

1. If $\nu_k^{1\mathrm{U}} \geq 0$, then it is optimal to serve the user;
2. If $\nu_k^{1\mathrm{U}} \leq 0$, then it is optimal to idle.

# Special Cases - 2U index

- Case of two users competing among themselves ($\beta = 1$)
- 2U index:

$$\nu_k^{\text{2U}} := \frac{c_k(\mu_k - \theta_k) + d_k \theta_k \mu_k}{\mu_k[1 - (1 - \mu_{3-k})(1 - \theta_k)]}.$$

- *Proposition:* Suppose $\nu_k^{1U} \geq 0$
  1. If $\nu_1^{\text{2U}} \geq \nu_2^{\text{2U}}$, then it is optimal to serve user $1$;
  2. If $\nu_1^{\text{2U}} \leq \nu_2^{\text{2U}}$, then it is optimal to serve user $2$.
  3. It is optimal to idle if and only if $\nu_1^{1U} = \nu_2^{1U} = 0$
- For more users too technical to be solved

# Whittle's Relaxation (1988)

- Relax the sample path constraint: serve $1$ user on $\beta$-average

$$\left[\sum_{k\in\mathcal{K}} W_{k,X_k(t)}^{a_k(t)}\right] = 1 \Rightarrow \mathbb{B}_0^{\boldsymbol{\pi}}\left[\sum_{k\in\mathcal{K}} W_{k,X_k(\cdot)}^{a_k(\cdot)}\right] = 1$$

- We obtain the relaxed problem:

$$\max_{\boldsymbol{\pi}\in\Pi_{\boldsymbol{X},\boldsymbol{a}}} \mathbb{B}_0^{\boldsymbol{\pi}}\left[\sum_{k\in\mathcal{K}} R_{k,X_k(\cdot)}^{a_k(\cdot)}\right]$$

$$\text{subject to } \mathbb{B}_0^{\boldsymbol{\pi}}\left[\sum_{k\in\mathcal{K}} W_{k,X_k(\cdot)}^{a_k(\cdot)}\right] = 1.$$

# Lagrangian relaxation and decomposition

- Following Lagrange relaxation with multiplier $\nu$

$$\max_{\boldsymbol{\pi}\in\Pi_{\boldsymbol{X},\boldsymbol{a}}} \mathbb{B}_0^{\boldsymbol{\pi}}\left[\sum_{k\in\mathcal{K}} R_{k,X_k(\cdot)}^{a_k(\cdot)} - \nu\sum_{k\in\mathcal{K}} W_{k,X_k(\cdot)}^{a_k(\cdot)}\right] + \nu.$$

- Decomposing into $k$-User Subproblem due to independence

$$\max_{\widetilde{\pi}_k\in\Pi_{\boldsymbol{X},a_k}} \mathbb{B}_0^{\widetilde{\pi}_k}\left[R_{k,X_k(\cdot)}^{a_k(\cdot)} - \nu W_{k,X_k(\cdot)}^{a_k(\cdot)}\right].$$

# Solution - AJN index

- For user $k$, $\nu_{k,0}^{\mathsf{AJN}} := 0$ and

$$\nu_{k,1}^{\mathsf{AJN}} := \frac{c_k(\mu_k - \theta_k) + d_k\theta_k(1 - \beta + \beta\mu_k)}{1 - \beta + \beta\theta_k},$$

and for idling, $\nu_{K,0}^{\mathsf{AJN}} := 0$.

- *Proposition:* Suppose $\nu_k^{1U}$
  - if $\nu \leq \nu_{k,1}^{\mathsf{AJN}}$, then it is optimal to serve waiting user $k$;
  - if $\nu \geq \nu_{k,1}^{\mathsf{AJN}}$, then it is optimal not to serve waiting user $k$;

# AJN Rule for Original problem

- Feasible policy for the original problem constructed by optimal solution of the relaxed problem

- AJN rule: allocates service at time $t$ to job $k^*(t)$ such that:

$$k^*(t) \in \arg\max_{k \in \boldsymbol{K}} \nu^{AJN}_{k,X_k(t)}$$

- Heuristic rule

- Not necessarily optimal for the original problem

## Limiting cases of AJN index

- If $\theta_k = 0$ reduces to $c\mu$-rule

$$\nu_{k,1}^{\mathsf{AJN}} := \frac{c_k \mu_k}{1-\beta},$$

- Time-average version of the AJN index $(\beta = 1)$:

$$\nu_{k,1}^{\mathsf{AJN}} := \frac{c_k(\mu_k - \theta_k) + d_k \theta_k \mu_k}{\theta_k},$$

- Myopic version of the AJN index $(\beta = 0)$:

$$\nu_{k,1}^{\mathsf{AJN}} := c_k(\mu_k - \theta_k) + d_k \theta_k.$$

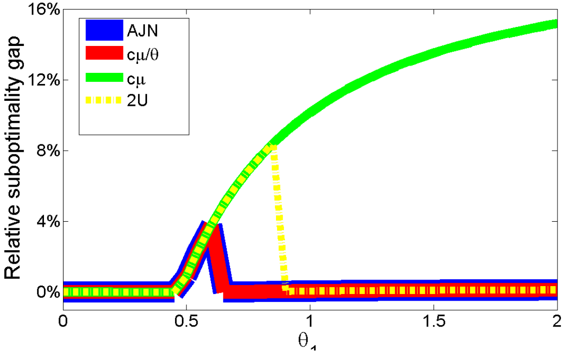
# Computational Experiments - Setting

- For higher relevance in applications:
  - investigating time-average performance
  - continuous-time model
- Two classes, each characterized by:
  - exponential rates: $\mu$, $\theta$ and Poisson arrivals $\lambda$
  - costs: $c$ and $d$
- Comparing rules: AJN, $c\mu/\theta, c\mu$ and 2U

$$\frac{c\mu}{\theta} := \frac{c_k\mu_k + d_k\theta_k\mu_k}{\theta_k} \text{ (Atar et al. 2010)}$$

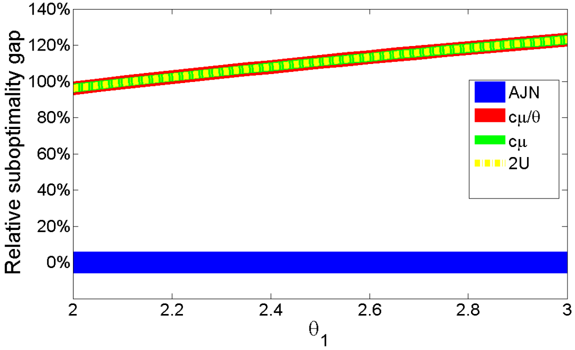
- We investigated a wide range of settings for the parameters in around 200 scenarios

# Computational Experiments - Scenario 1



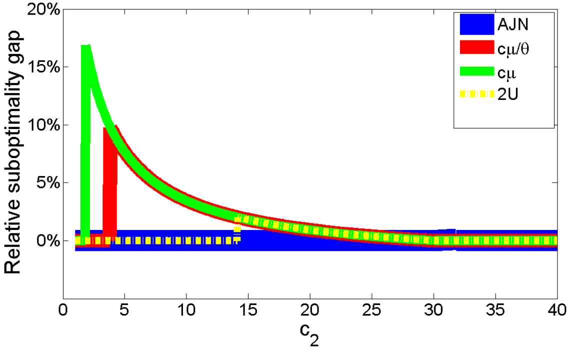

Setting: $\mu_1 = 0.7$ , $\mu_2 = 0.3$ , $\theta_2 = 0.2$ and
$c_1 = c_2 = d_1 = d_2 = \lambda_1 = \lambda_2 = 1$

# Computational Experiments - Scenario 2




Setting: $\mu_1 = 0.4$ , $\mu_2 = 0.59$ , $\theta_2 = 4$ and $c_1 = c_2 = d_1 = d_2 = \lambda_1 = \lambda_2 = 1$

# Computational Experiments - Scenario 3




Setting: $\mu_1 = 0.4$ , $\mu_2 = 0.22$ , $\theta_1 = 0.1$, $\theta_2 = 0.2$ and

$$c_1 = d_1 = d_2 = \lambda_1 = \lambda_2 = 1$$

# Conclusion

- We investigated the problem of job scheduling with user abandonments
- For the problem with one or two users, we obtained optimal solutions (indices: 1U and 2U)
- AJN index - optimal solution of relaxed problem
- AJN-rule performance:
  - AJN is often optimal, if not - suboptimality small
  - In most cases the AJN-rule outperforms the $c\mu/\theta$ and $c\mu$
  - AJN's biggest improvement is when it is optimal to idle
- Further work
  - Determine under what conditions the AJN-rule is optimal

# Thank you for your attention

- Acknowledgement


  - "ZA EFM" Alumni association
  - Czechoslovakia section of IEEE
  - Literary fund of Slovakia
  - Research partially supported by grant MTM2010-17405 of the MICINN (Spain) and grant PI2010-2 of the Department of Education and Research (Basque Government).