Takmer-Optimálne Riešenie Stochastického a Dynamického Problému Využitia Vzácnych Zdrojov

Peter Jacko*

December 17, 2009

*Basque Center for Applied Mathematics, Spain

Kto, Kde a Kedy

- UPJŠ, prof. Jendrol'
  - ▷ Bc. matematika, 2002
  - ▷ Mgr. Matematický manažment, 2003
- Universidad Carlos III de Madrid, Spain
  - ▷ Erasmus, 2002
  - ▷ D.E.A. (Master) Štatistika a operačný výskum, 2006
  - ▷ Ph.D. Business Administration and Quantitative Methods, 2009
- Basque Center for Applied Mathematics (BCAM)
  - ▷ post-doc

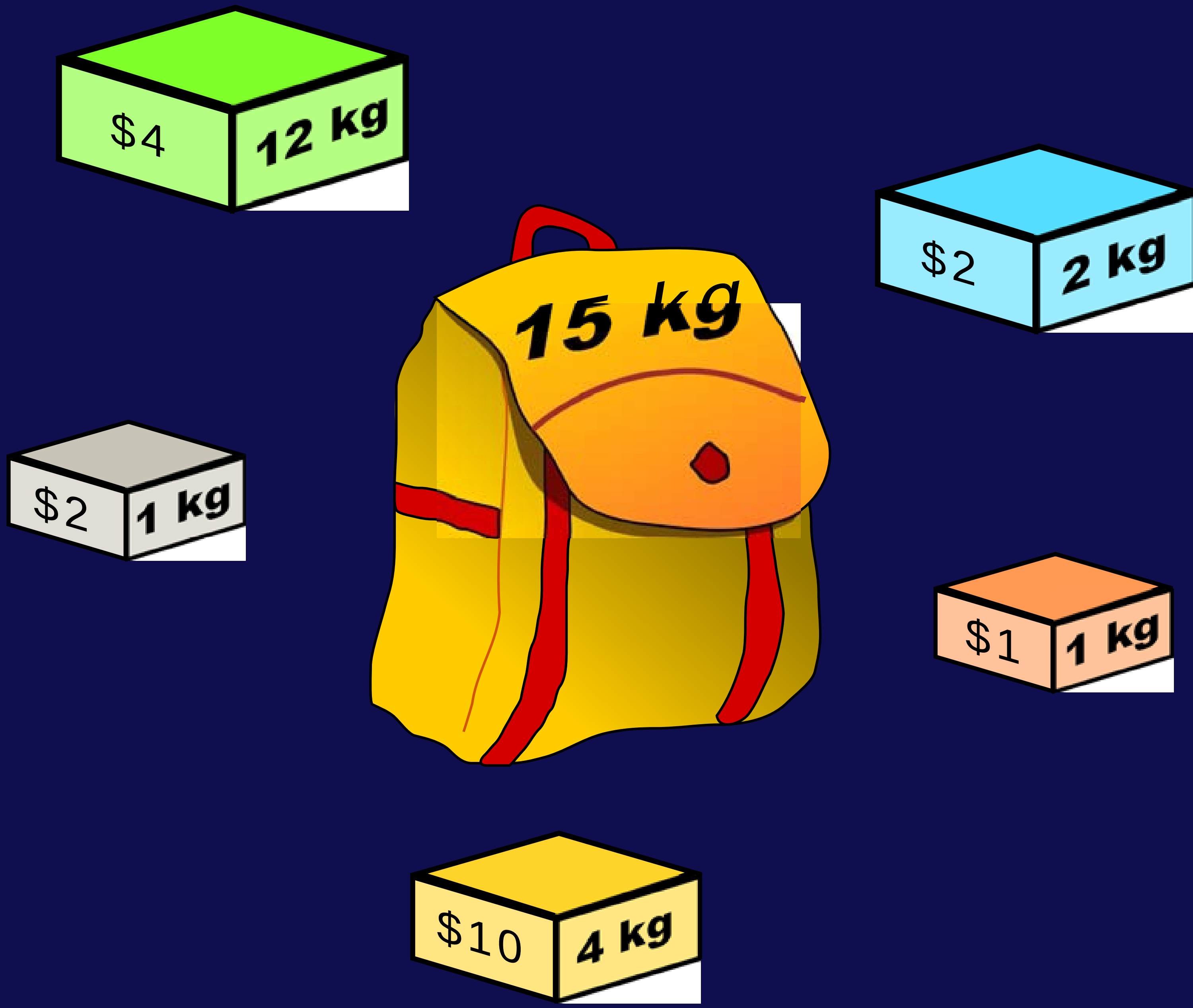
Resource Allocation in Telecommunications

- Congestion control
  - ▷ maximize throughput (choose preferred flows)
- Routing
  - ▷ minimize packet losses (choose preferred paths)
- Admission control
  - ▷ minimize delays (choose preferred packets)
- Fairness
  - ▷ maximize users' utilities (choose preferred users)

Resource Allocation in Robotics

- Arises within reinforcement learning
- Behavior coordination & navigation in animats
  - ▷ maximize utility (choose preferred behavior)
- Ranking in web search robots
  - ▷ minimize searching time (choose preferred document)
- Multi-target tracking & environment mapping
  - ▷ maximize map correctness (choose preferred object)

Static Case: Knapsack Problem



Resource Allocation Problem

- Stochastic and dynamic
- There are a number of independent competitors
- Constraint: resource capacity
- 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
- How to design a good dynamic priority rule?

Stochastic Programming Framework

- Stochastic programming = Markov Decision Processes
- Discrete time model ($t = 0, 1, 2, \dots$)
- Competitor $k \in \mathcal{K}$ is defined by
  - ▷ state space $\mathcal{N}_k$, action space $\mathcal{A}$
  - ▷ 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 $X_k(t)$ and action process $a_k(t)$

Example: 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 jobs k
  - ▷ μ_k = service rate for jobs k

- $\mathcal{N}_k := \{\text{'completed'}, \text{'waiting'}\}$, $\mathcal{A}_k := \{\text{'serve'}, \text{'wait'}\}$

- expected one-period capacity consumption

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

Example: Job Sequencing Problem

- expected one-period reward

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

- one-period transition probability matrices

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

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

Resource Allocation Problem

- Formulation under the β-discounted criterion:

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

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

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

Whittle's Relaxation

- Fill the capacity in expectation
  - ▷ infinite number of constraints is replaced by one

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

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

- Provides an upper bound for RAP

Lagrangian Relaxation

- Pay cost λ for using the capacity
  - ▷ the constraint is moved into the objective

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

- Also provides an upper bound for RAP
- This decomposes due to competitor's independence into single-competitor subproblems (easier to solve)

Priority Rules

- Assign to each competitor a dynamic price
  - ▷ the rule at each moment is to be greedy: prefer competitors with higher current prices
- The prices arise in the solution of the subproblem
  - ▷ when λ is small, it is optimal to use all the capacity
  - ▷ when λ is large, it is optimal to use no capacity
  - ▷ prices are values of λ when optimal solution changes
  - ▷ optimal policy: use capacity iff price lower than λ
- In general, this gives nearly-optimal solution to RAP

Optimality of Priority Rules

- Often in problems with symmetric competitors
- E.g.: in routing to parallel queues, route to:
  - ▷ the Shortest Queue (if min. delays)
- E.g.: sequencing of customers to service:
  - ▷ the Shortest Service Time (if min. waiting time)
  - ▷ the Least Empty Buffer Space (if min. losses)
  - ▷ the Shortest Queue (if min. delays)
- These values are the "prices"

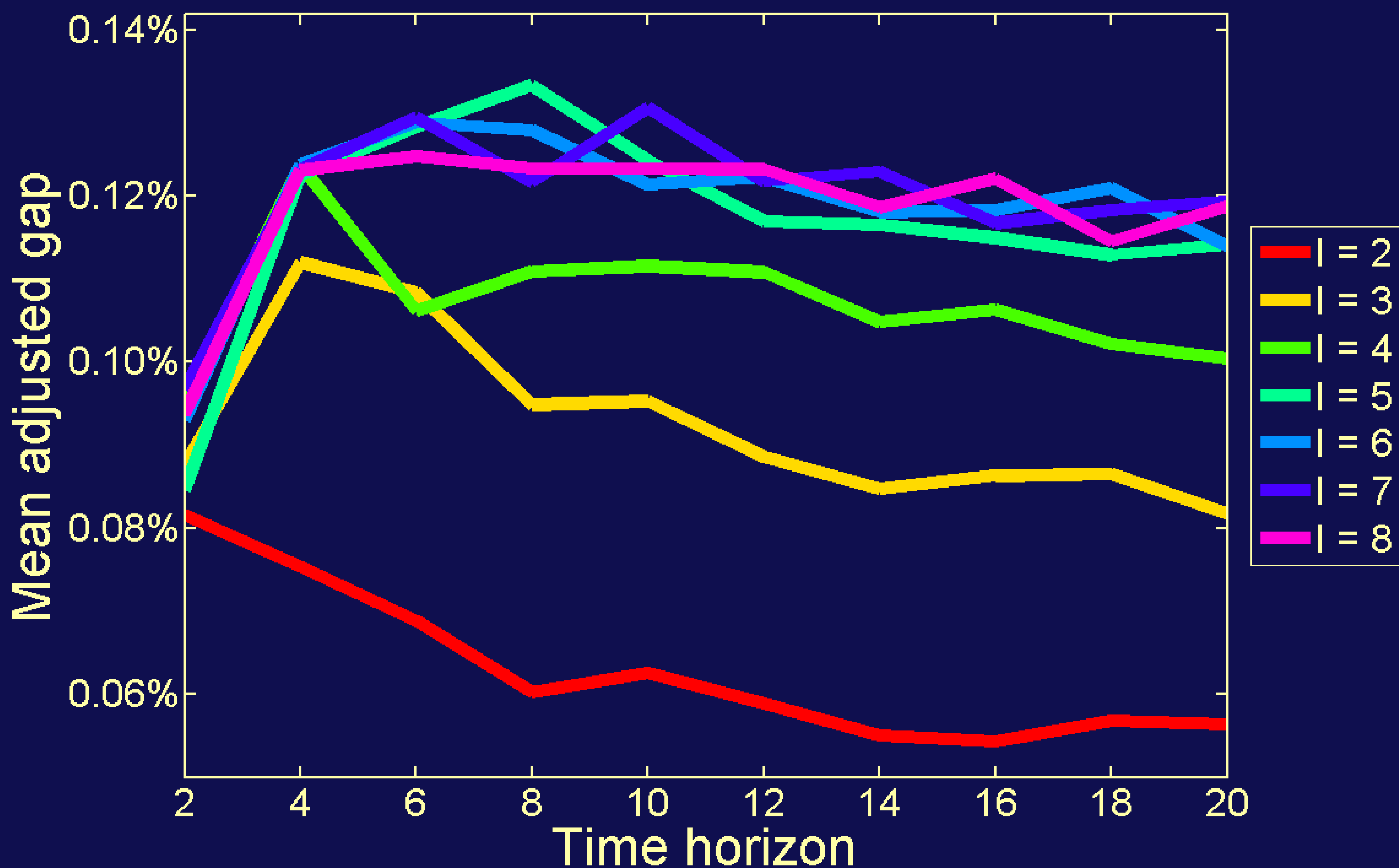
Optimality of Priority Rules

- Also in asymmetric problems with simple dynamics
- $c\mu$-rule (Cox & Smith '61)
  - ▷ several classes $k \in \mathcal{K}$ of arriving customers/packets
  - ▷ c_k = cost of waiting for class k
  - ▷ μ_k = service rate for class k
  - ▷ assigning priority to the class k with largest $c_k \mu_k$
  - ▷ optimal in several simple resource allocation problems
- Gittins index rule ('72) for multi-armed bandit problem
- Klimov index rule ('74) for $M/G/1$ model with feedback

Priority Rules in More Complex Problems

- e.g.: Niño-Mora ('02)
  - ▷ routing to parallel queues with asymmetric waiting costs and service rates
- e.g.: Jacko (Ph.D. thesis '09)
  - ▷ routing to parallel queues with belated information
  - ▷ congestion control for TCP flows
- Price computation:
  - ▷ in general, by an algorithm in at most cubic time
  - ▷ after an analysis, sometimes obtained in a closed form

Example: Performance of a Priority Rule



Thank you for your attention

Problems in Telecommunications

- Fall into resource allocation problems
- Advantages:
  - ▷ decentralized control
  - ▷ natural for creating priority tables
  - ▷ dynamic prices yield structural results
  - ▷ nearly-optimal (optimal in expectation)
- Disadvantages
  - ▷ prices may not exist

Problems in Telecommunications

- However, dynamic prices are scarcely used
  - ▷ Niño-Mora '02, '06
  - ▷ Raissi-Dehkordi & Baras '02
  - ▷ Goyal et al. '06
  - ▷ Jacko '09
- Optimality of ad-hoc priority rules is usually analyzed
  - ▷ Glazebrook et al. '04, '04, '07
  - ▷ Ehsan & Liu '04, '05, '06, '07

Problems in Telecommunications

- Niño-Mora '02:
  - ▷ queues with finite buffers
  - ▷ consider a rejection cost as the wage
  - ▷ assume concave nondecreasing service rates
  - ▷ assume convex nondecreasing holding costs
  - ▷ price-based characterization of optimal threshold policy
  - ▷ as rejection cost grows, start rejecting customers under longer queue
  - ▷ priority rule heuristic for routing to parallel queues

Problems in Telecommunications

- Niño-Mora '06:
  - ▷ queues with finite buffers
  - ▷ analyzes loss-sensitive and delay-sensitive queues
  - ▷ rejecting cost vs. discounted holding-forever cost
  - ▷ loss-sensitive: fewest-empty-buffer-spaces rule
  - ▷ delay-sensitive: shorter-queue rule
  - ▷ both converge to $c\mu$ rule for infinite buffers
  - ▷ throughput maximization is special case

Problems in Telecommunications

- Goyal, Kumar & Sharma '06:
  - ▷ transmissions over polled multiaccess fading channel
  - ▷ voice $\succ$ streaming media $\succ$ files
  - ▷ infinite buffers, delayed information
  - ▷ poll-and-response system
- Raissi-Dehkordi & Baras '02:
  - ▷ pulling broadcast scheduling (teletext with feedback)
  - ▷ minimize weighted average waiting time

Problems in Telecommunications

- Glazebrook et al. '04:
  - ▷ server allocation to impatient (perishable) tasks
  - ▷ reduces to Gittins indices in a special case
- Glazebrook & Kirkbride '04:
  - ▷ routing of background jobs in distributed PC systems
  - ▷ ad-hoc prices (static policy improvement)
- Glazebrook & Kirkbride '07:
  - ▷ routing to heterogeneous unreliable servers
  - ▷ ad-hoc prices (DP policy improvement)

Problems in Telecommunications

- Ehsan & Liu '04, '05, '06, '07:
  - ▷ wireless server allocation with delays
  - ▷ minimize expected holding costs
  - ▷ ad-hoc prices (myopic)
  - ▷ give sufficient optimality conditions (special cases)