Effective computation of the critical levels in a stochastic inventory control

J. Bollati A. Bušić E. Hyon

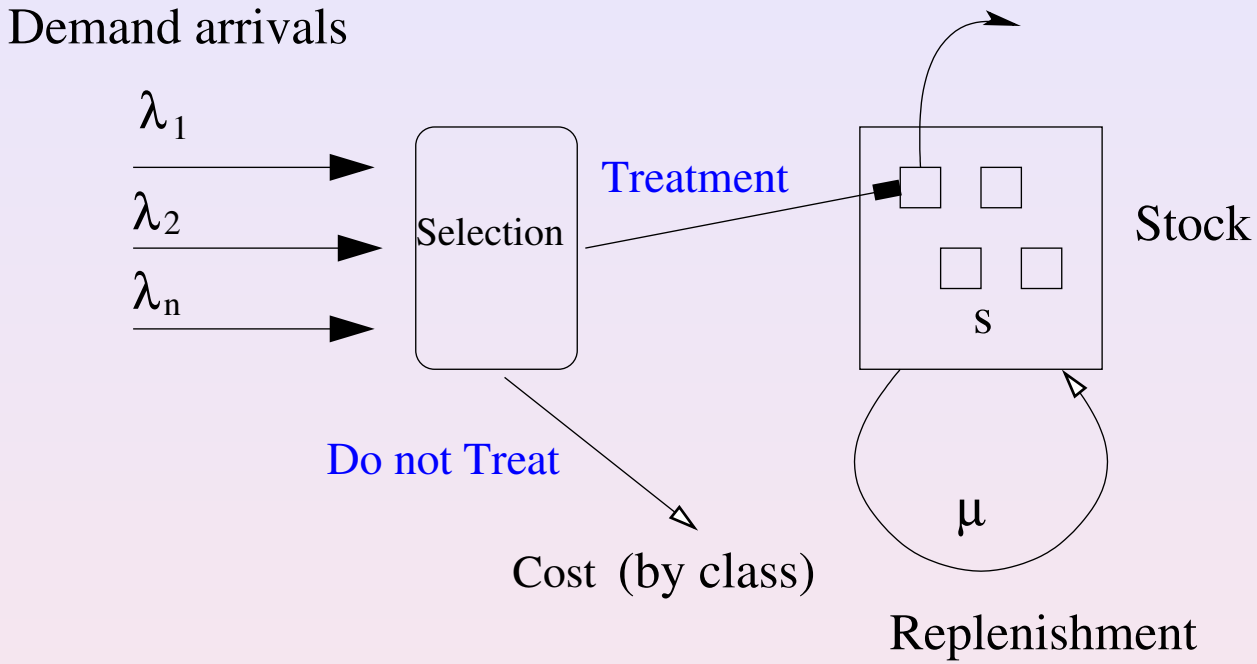
Universidad Nacional de Rosario, INRIA, LIP6

Juillet 2014
Grenoble

Outline

Progress

An inventory control



Customer parameters

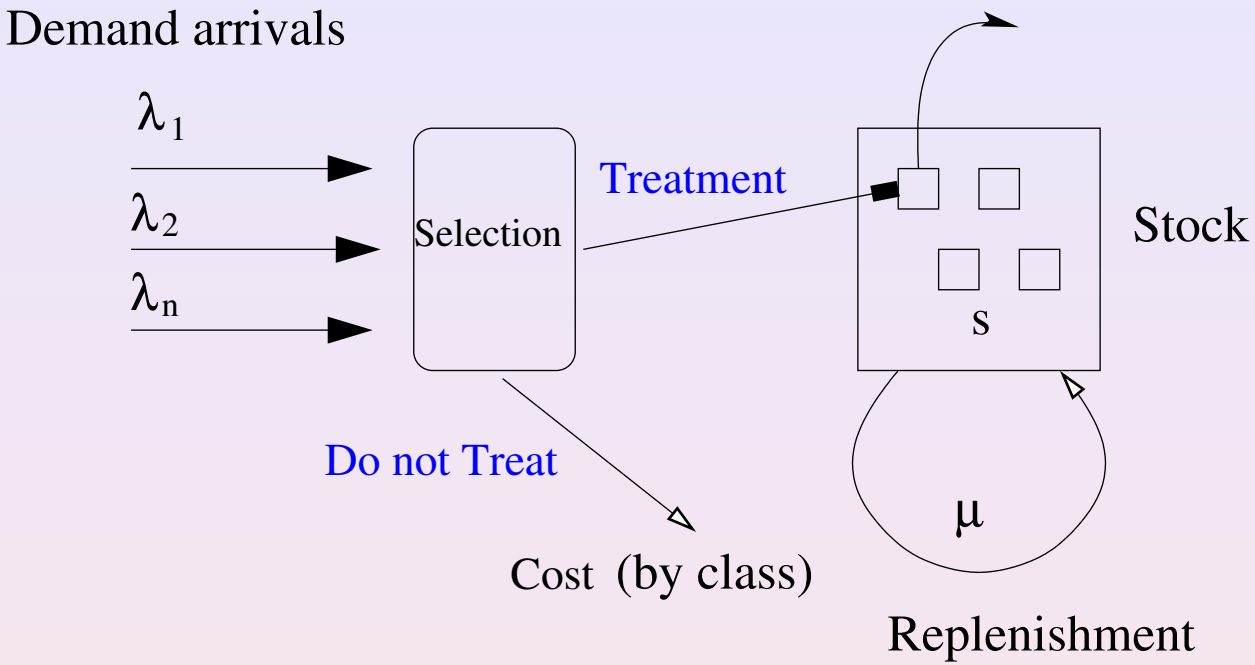
J classes :

- c_j rejection cost
- λ_j arrival rate Poisson

Stock parameters

- Stock size S
- replenishments are exponential
- μ is the replenishment parameter

An inventory control

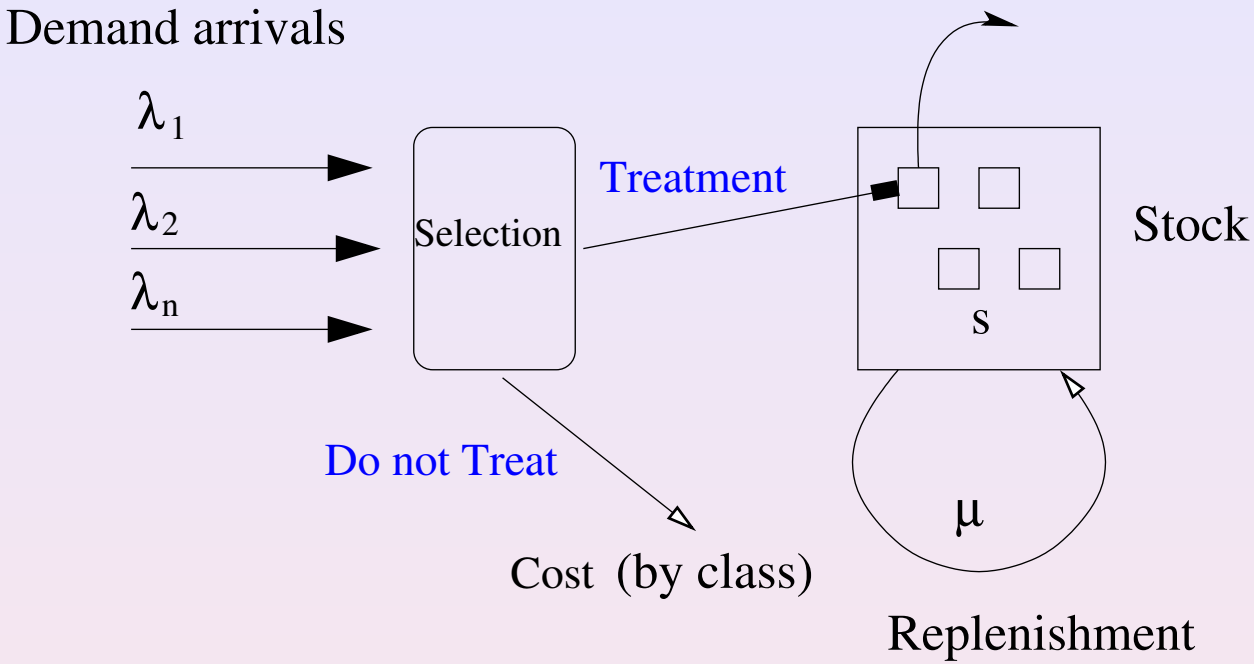


Problem

Customers haven't the same cost ($c_1 > c_2 > \ldots > c_J$).

- Satisfaction of demands of all customers indifferently lead to a full system.
- Full system means no treatment of customer with the highest cost

An inventory control



Problem

Customers haven't the same cost ($c_1 > c_2 > \ldots > c_J$).

- Satisfaction of demands of all customers indifferently lead to a full system.
- Full system means no treatment of customer with the highest cost

We want to choose the customer to treat

Finite size or holding Cost

In the literature (e.g. *Vericourt 02*) two models

1. with backlog
2. with rejection

In the literature (*Pouters 02* for an overview) two main models :

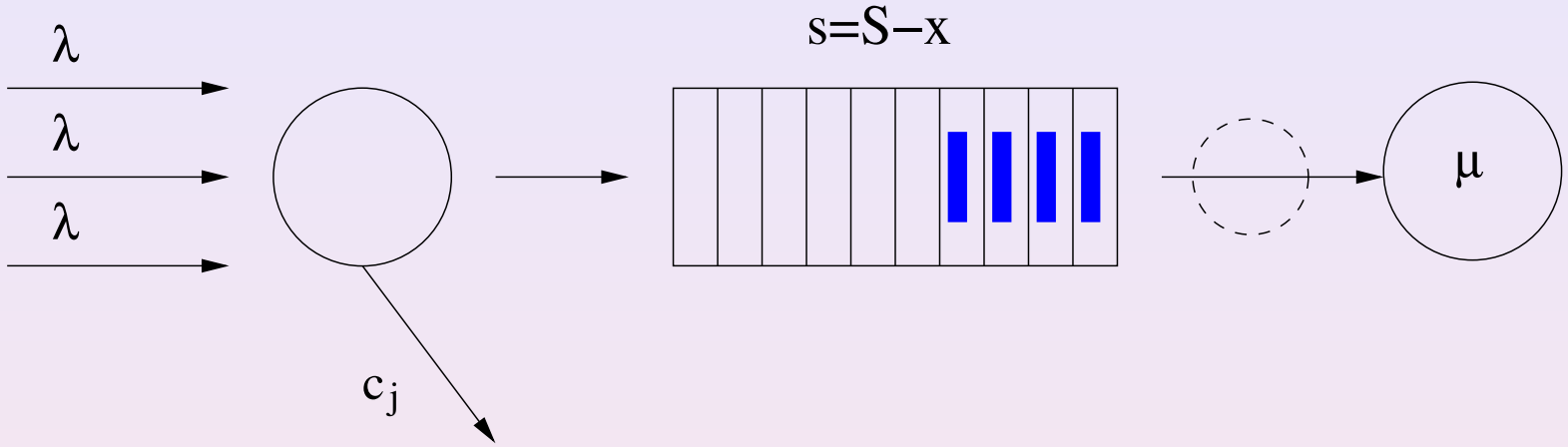
1. with Finite Stock
2. with Holding Costs (per capita and per time unit)

We can with from model two to model one easily.

Here we assume a finite stock size AND an holding cost.

Representation using a queue

We consider items in replenishment instead of the inventory.



Thus the previous model can be represented by a queue : jobs in the queue represents the item in replenishment.

- When the queue is empty then the stock is full (S items in the stock).
- When the queue is full the stock is empty.
- If a demands is satisfied then the item is immediately admitted in replenishment and a job enters in the queue :
  This is an admission control policy.

Policy

Policy

We define a policy π as a sequence of decision rules

$$\pi = (\pi_1(h_1), \ldots, \pi_n(h_n), \ldots).$$

A decision rule maps the history h (states of the process, events, decision of the controller) to an action (here acceptance or rejection).

A policy π is a Markov stationary deterministic policy *i.e.* if $\pi^* = (\pi(x), \pi(x), \ldots)$.

Dynamical behaviour of the system

Once the policy is fixed the dynamic of the system is random :

- T_n^π epoch of the n^{th} transition
- x_n^π state just after the n^{th} transition.

Objective function

Average cost

We define $v_t(x)$ as

$$V_t^\pi(x) = \mathbb{E}\Big[\sum_{n=0}^{\nu_t-1}(T_{n+1}^\pi - T_n^\pi)c_R(x_n^\pi) + \sum_{n=0}^{\nu_t-1} c_I(x_n^\pi, \pi(x_n^\pi))\Big]. \tag{1}$$

with ν_t the number of transitions until t ;
c_R the rate cost ;
c_I the lump cost.

The average cost is ρ^π :

$$\rho^\pi = \lim \inf_{t\to\infty} \frac{1}{t} V_t^\pi(x).$$

We want to find $\rho^* = \min_\pi \rho^\pi$.

This means to characterize the optimal decision rule $\pi^*(x)$ w.r.t the state.

N-stage total cost

Assume a policy of length N : $\pi(N)$. The minimal expected N-stage total cost is

$$V_N^\pi(x) = \min_{\pi(N)} E_x^\pi \left[\sum_{n=0}^{N-1} \left((T_{n+1}^\pi - T_n^\pi) c_R(x_n^\pi) + c_I(x_n^\pi, \pi(x_n^\pi)) \right) \right] \tag{2}$$

Relation with the average cost

From Puterman an alternate definition of ρ^π is

$$\rho^\pi = \liminf_{N\to\infty} \frac{\mathbb{E}\left[\sum_{n=0}^{N} \left((T_{n+1}^\pi - T_n^\pi) c_R(x_n^\pi) + c_I(x_{T_n^\pi}^\pi, \pi(x_{T_n^\pi}^\pi)) \right)\right]}{\mathbb{E}\sum_{n=0}^{N} (T_{n+1}^\pi - T_n^\pi)} .$$

When the process is uniformized then

$$\lim_{N\to\infty} \frac{1}{N} V_N^\pi = \frac{\rho^\pi}{\Lambda} .$$

Dynamic Programming

Puterman[96] approach : transformation into a fix point equation

We transform the Equation (1) into a fix point Equation : the *dynamic programming* equation or *Bellman Equation*.

$$V(x) + \rho^{\pi} = \min_{q \in \text{Action Set}} c(x,q) + \sum_{y} \mathbb{P}(y|(x,q)V(y). \tag{3}$$

This fix point equation has one unique solution V^*.

Koole [06] approach

We transform (2) under $V_N = T(V_{N-1})$,
with

$$T(V)(x) = \frac{1}{\Lambda}\left(\bar{h}(x) + \mu v((x-1)^+) + \sum_{j=1}^{J} \lambda_j \min\{c_j + V(x), V(x+1)\}\right) \tag{4}$$

We have $\lim_{N\to\infty} \frac{1}{N} V_N = \frac{\rho^{\pi}}{\Lambda}$

Numerical computation

We want

We want the couple (ρ^*, π^*)

- the optimal value ρ^*
- The optimal policy is $\pi^*(x) = \arg\min c(x, q) + \mathbb{E}_x^q(V)$

Value iteration

1. Both Equation 3 and 2 can be expressed under $V_n = TV_{n-1}$
2. We start with $V_0 = 0$
3. until

$$\max_x (V_n(x) - V_{n-1}(x)) - \min_x (V_n(x) - V_{n-1}(x)) \leqslant \varepsilon$$

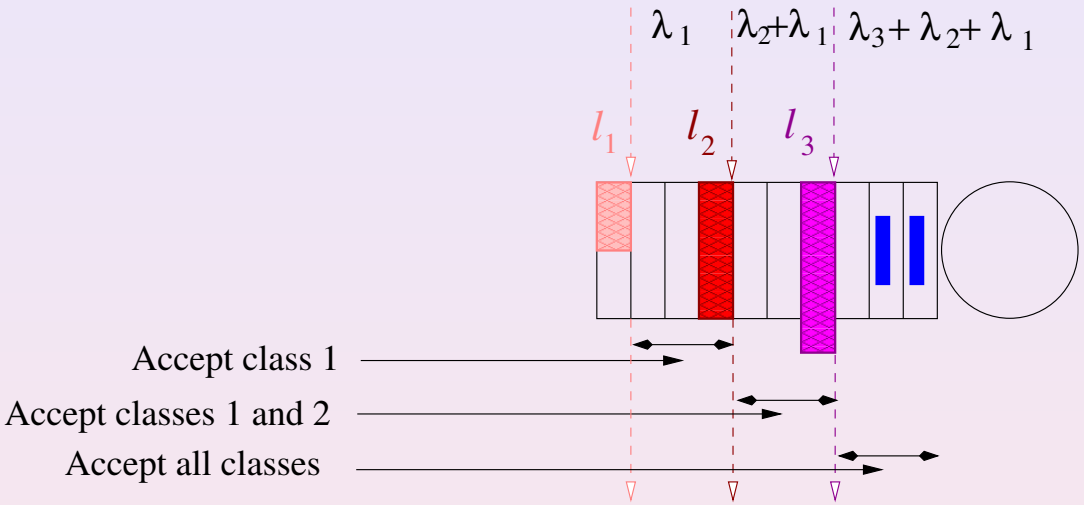
But :

Problem : the curve of dimensionality

Progress

Critical levels policy (*threshold* or *control limit*)

We work with with policy with a special form : The critical levels policy : The acceptation depends on a *level* in the queue load.



Literature review for the example

- Ha 97 with exponential replenishment
- Ha 2000 with Erlang Replenishment
- Wieczorek, Busic and H 2011 with Hypo exponential

In the Literature

Axis 1 Optimality proofs

- Show that thresholds policies are optimal.
- Classical framework Koole(2006), Glasserman & Yao(1994)

Formal Approach (sketch)

- Showing that when the value functions is convex it implies a threshold.

$$V(x+1) - V(x) \geqslant c_j$$

- Showing the property is kept through Fix Point Equation.

In the Literature

Axis 2 Effective Computation of the Levels

- No framework no generic method
- Only ad hoc methods
- Very few proofs about the optimality
- Very few Stochastic methods such as stochastic gradient.

Usual approach

Computing the value of V_l for a fixed l by

1. Classical BD process methods
2. Markov Reward Process

This gives a deterministic non linear integer problem

The aim is then to determine the set of optimal values of the levels.

Progress

Expression in a deterministic problem (Dekker 97)

MC dynamic

- Once any stationary deterministic policy is applied, the behaviour of the system is described by a stochastic process.
- The behaviour of the system is described by a Markov Chain
- Here we have an $M/M/1/S$ queue with an arrival Poisson process with variable rates
- We can compute stationary probability and deduce the average cost V_l.

Deducing average cost

- Computation of the stationary probabilities ($q_k(l)$ the probability to have k customers when the level is l)
- Computation of the *fill rate*
  $\beta_j(l) = \sum_{k=0}^{l_j-1} q_k$ the proportion of items j that are delivered
- The average cost q.r.t vector l is

$$V_l = \sum_j \lambda_j c_j (1 - \beta_j(l))$$

A Minimization Problem

We want

$$\text{Minimize } \rho_l = \sum_{j \in J} \left(c_j \lambda_j \sum_{k=l_j}^{S} q_k(l) \right) + \sum_{k \in S} \bar{h}(k) q_k(l)$$

$$\text{subject to } 0 \leqslant l_J \leqslant l_{J-1} \leqslant \ldots \leqslant l_1 \leqslant S \ .$$

Local search

Dekker (1997) and van Houtum (2002)

Model a bit slightly different

- The size S is variable
- There is an holding cost

Propose an exact method to determine a value of S
Propose 3 heuristics of local search to compute the optimal levels for a fixed S.

A local search

while Modification of the threshold occur **do**
  for J from 1 to J **do**
    if $V(I + e_j) \leqslant V(I)$ **then**
      $I = I + e_j$
    end if
  end for
end while

Greedy

Model of backlogging

Model is a bit slighty different :

No losses but backlog of unsatisfied demands ($\approx$ infinite queue)

The goal : to find the levels of backlog..

Vericourt (2002) exponential services, Karaesmen (2009) Erlang services

Greedy approach

- Decomposition of the problem w.r.t. the problem with smaller size. $P(n) = f(\lambda_n, c_n, P(n-1))$.
- Greedy approach :
  - sorting classes by decreasing order of the backlog cost.
  - Optimize class k (with $1 \leqslant k \leqslant n$) by considering that classes of greater index than k are not present
- Greedy approach is optimal when backlogging.
- Close formula for a single threshold (old result) in exponential case.

Branch and Bound Method

Hard to prove convexity

Convexity with respect to l *i.e.* $l \mapsto V_l(x)$

$$V_l(x) - V_{l-1}(x) \leqslant V_{l+1}(x) - V_l(x)$$

Branch and Bound

- It exists lower and upper bound by merging classes and solving a smaller problem.
- But this induces an order of the levels to branch.

Lower Bound

If I search a lower bound according to a level of index k.
I merge all the customers with class $j \geqslant k$ by fixing there cost to c_J.

$$V_{(l_1,l_2,\ldots,l_J)}(x) \geqslant V_{(l_1,l_2,\ldots,l_{k+1})}$$

Continuous Approximation

An acceptance ratio

- We consider we have now an acceptance rate α_j instead of a level l_j for each class.
- We accept a class j customer with a Bernoulli probability α_j.
- $l_j = \lceil S\alpha_j \rceil$
- We have an $M/M/1/S$ queue with a fixed rate Poisson Process $\lambda(\alpha) = \sum_j \lambda_j \alpha_j$

Cost Function

Once the vector $\alpha = (\alpha_1, \ldots, \alpha_J)$. The probability q_k depends on α (so denoted by $q_k(\alpha)$)

$$\rho_{app} = \sum_j \alpha_j \lambda_j c_j + q_S(\alpha) \sum_j \frac{\lambda_j}{\lambda} c_j$$

But : hard to prove the function is convex in l and gradient descent is not proved.

Progress

RunTime Comparizons

We run the different methods on 520 instances.
Applied for different sizes of the stock.

J	S	Exhaustif (ordre)	LS	G	VI	VI (ordre)	B&B
3	5	0.0031	0.0021	0.0017	0.0836	0.0758	0.0066
3	10	0.0117	0.0041	0.0039	0.50	0.4572	0.0203
3	50	0.647	0.0340	0.0500	18.10	16.581	0.8569
3	99	4.440	0.1062	0.1767	48.52	44.447	6.1171

Precision Comparizons

130 instances
Average relative Error

J	S	LS	G	VI	VI (ordre)	B&B
3	5	0	0. 009	$3.8\ 10^{-6}$	$3.84\ 10^{-6}$	0
3	10	0	0.014	$2.14\ 10^{-5}$	$2.14\ 10^{-5}$	0
3	50	0	$8.36\ 10^{-5}$	0.0033	0.0033	0
3	99	0	$4.15\ 10^{-6}$	0.0125	0,0125	0

Average relative error for instances for which the min is not reached

J	S	LS	G	VI	VI (ordre)	B&B
3	5	0	0.09 (13)	$2.94\ 10^{-5}$ (17)	$2.94\ 10^{-5}$ (17)	0
3	10	0	0.08 (24)	0.000147 (19)	0.000147 (19)	0
3	99	0	$4.5\ 10^{-5}$ (11)	0.346 (46)	0.346 (46)	0

Progress

Conclusion and perspective

Conclusion

- Brief review of Literature
- Short comparizon between methods

Perspective

- Enlarge the comparizons
- Comparizons using bounds
- Enlarge the methods of computation
  - SDDP and Stochastic Programming
  - Markov Reward Process

Markov Reward Process

In order to compute quickly lower bound.

MRP

- I assume a continuous time Markov Chain such that the transition times in all the state is bounded by Λ.
- We observe the continuous time process at epochs with interval time that follows an exponential with parameter Λ
- We capture transition epochs AND epochs with no transition.
- The kernel is then

$$P(x,y) = \begin{cases} \frac{Q(x,y)}{\Lambda} & \text{for } x \neq y, \\ 1 - \sum_{z \neq x} \frac{Q(x,z)}{\Lambda} & \text{for } x = y \end{cases}$$