Dynamic Resource Allocation via Objective Function Approximation

With Experiments on Queueing and (s,S) Inventory Problems

Guowei Sun[1] Dingyuan Xu[2]

[1]Department of Mathematics
University of Maryland

[2]Department of Civil Engineering
University of Maryland

Dec.11th / BMGT835 Course Project

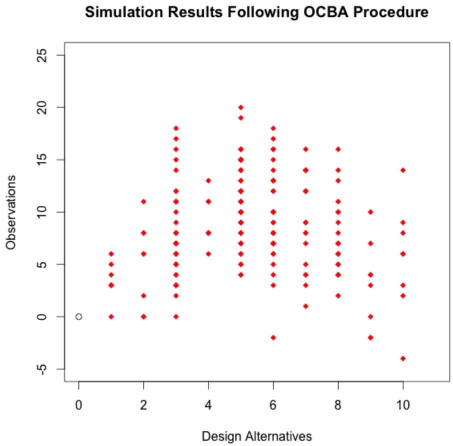
Outline

Outline

Motivation

Independence Between Alternatives in Ranking and Selection

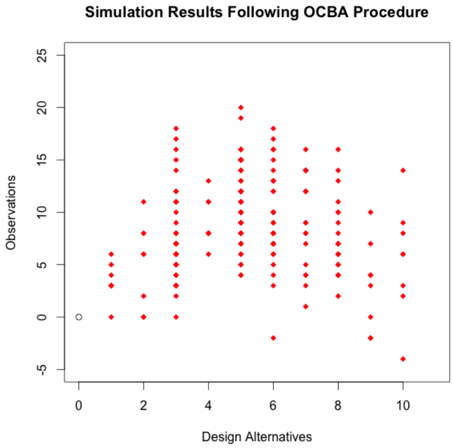
- We make observations (simulations) on all alternatives
- We allocate our simulation budget according to some rules
- Conclusions about performance of each alternatives is determined by the simulation results on this single alternative



Motivation

Independence Between Alternatives in Ranking and Selection

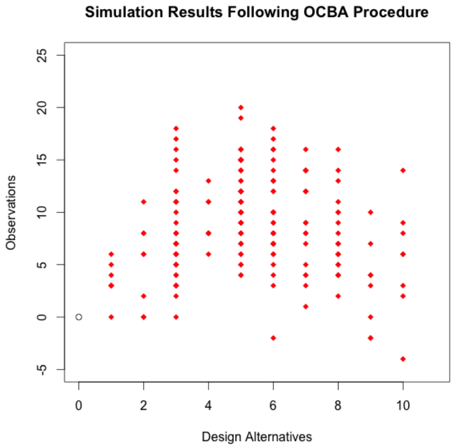
- We make observations (simulations) on all alternatives
- We allocate our simulation budget according to some rules
- Conclusions about performance of each alternatives is determined by the simulation results on this single alternative



Motivation

Independence Between Alternatives in Ranking and Selection

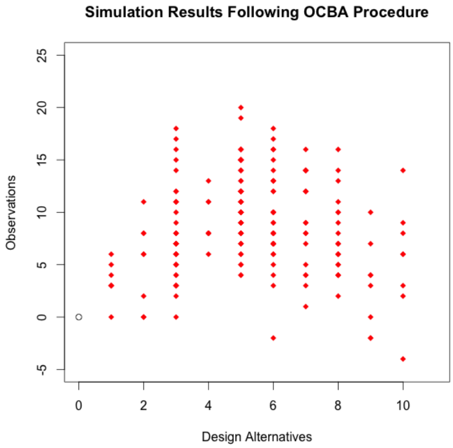
- We make observations (simulations) on all alternatives
- We allocate our simulation budget according to some rules
- Conclusions about performance of each alternatives is determined by the simulation results on this single alternative



Motivation

Independence Between Alternatives in Ranking and Selection

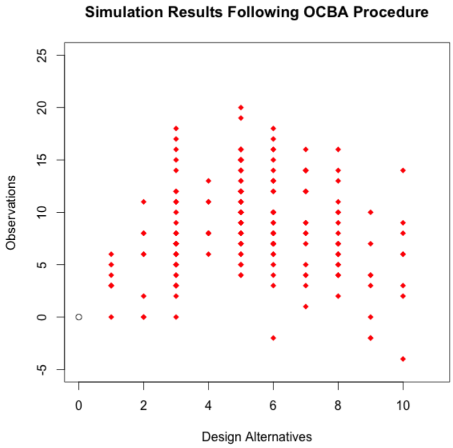
- We make observations (simulations) on all alternatives
- We allocate our simulation budget according to some rules
- Conclusions about performance of each alternatives is determined by the simulation results on this single alternative



Motivation

Independence Between Alternatives in Ranking and Selection

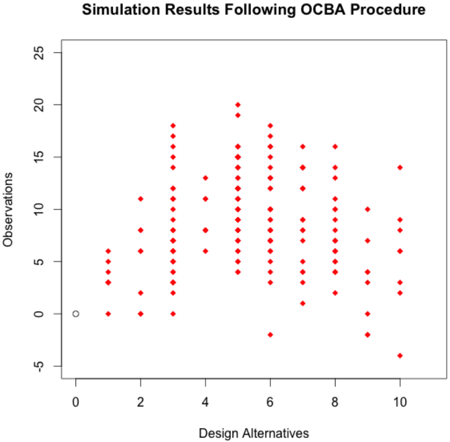
- We make observations (simulations) on all alternatives
- We allocate our simulation budget according to some rules
- Conclusions about performance of each alternatives is determined by the simulation results on this single alternative



Motivation

Independence Between Alternatives in Ranking and Selection

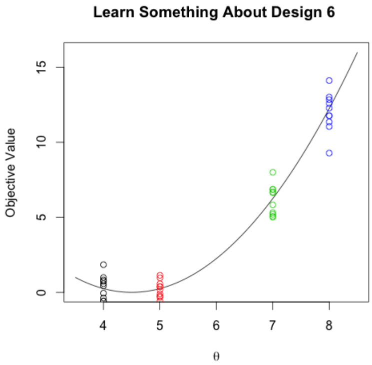
- We make observations (simulations) on all alternatives
- We allocate our simulation budget according to some rules
- Conclusions about performance of each alternatives is determined by the simulation results on this single alternative



Motivation

Objective Functions in Stochastic Approximation

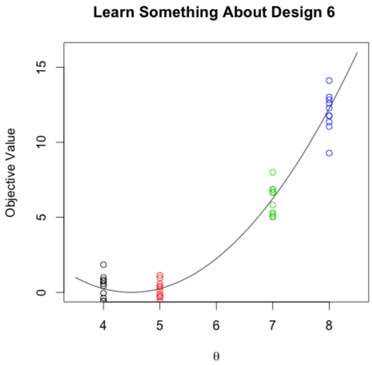
- In order to do stochastic approximation, we need very nice functions
  - No local minimums
  - Almost Everywhere Differentiable
- We should be able to learn *"something"* about alternative 6 from observations on alternatives 4,5,7,8



Motivation

Objective Functions in Stochastic Approximation

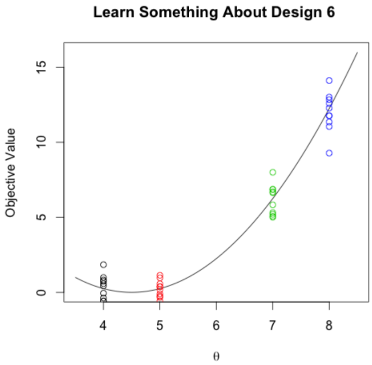
- In order to do stochastic approximation, we need very nice functions
  - No local minimums
  - Almost Everywhere differentiable
- We should be able to learn *"something"* about alternative 6 from observations on alternatives 4,5,7,8



Motivation

Objective Functions in Stochastic Approximation

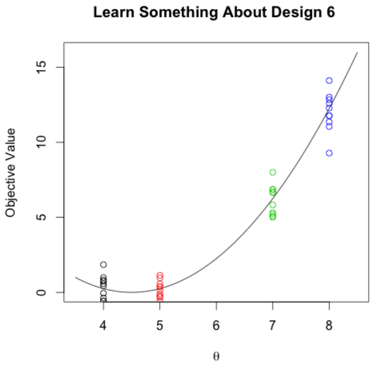
- In order to do stochastic approximation, we need very nice functions
    - No local minimums
    - Almost Everywhere differentiable
- We should be able to learn *"something"* about alternative 6 from observations on alternatives 4,5,7,8



Motivation

Objective Functions in Stochastic Approximation

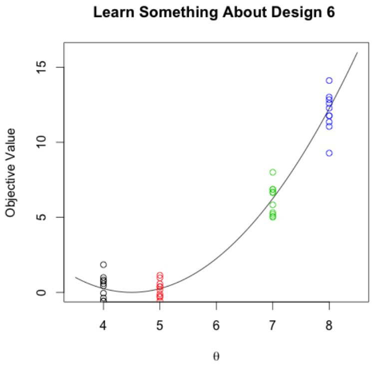
- In order to do stochastic approximation, we need very nice functions
  - No local minimums
  - Almost Everywhere differentiable
- We should be able to learn *"something"* about alternative 6 from observations on alternatives 4,5,7,8



Motivation

Objective Functions in Stochastic Approximation

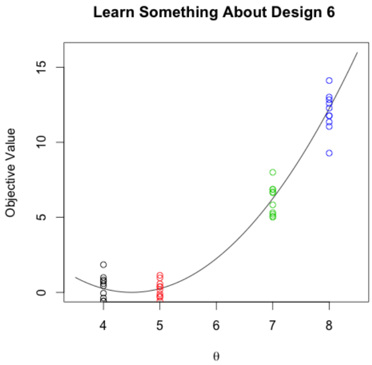
- In order to do stochastic approximation, we need very nice functions
    - No local minimums
    - Almost Everywhere differentiable
- We should be able to learn *"something"* about alternative 6 from observations on alternatives 4,5,7,8



Motivation

Objective Functions in Stochastic Approximation

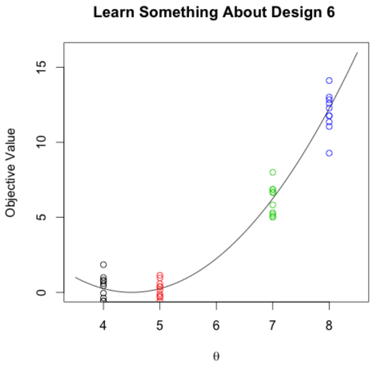
- In order to do stochastic approximation, we need very nice functions
  - No local minimums
  - Almost Everywhere differentiable
- We should be able to learn *"something"* about alternative 6 from observations on alternatives 4,5,7,8



Motivation

Objective Functions in Stochastic Approximation

- In order to do stochastic approximation, we need very nice functions
  - No local minimums
  - Almost Everywhere differentiable
- We should be able to learn *"something"* about alternative 6 from observations on alternatives 4,5,7,8



Motivation

Make Use of All Available Information?

Objective Function Approximation

Estimate The Objective Function

If we are dealing with the problem

$$\max_{\theta\in\Theta} J(\theta)$$
$$J(\theta) = E[T(\theta)]$$
$$\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$$

With Observations on Each Alternative

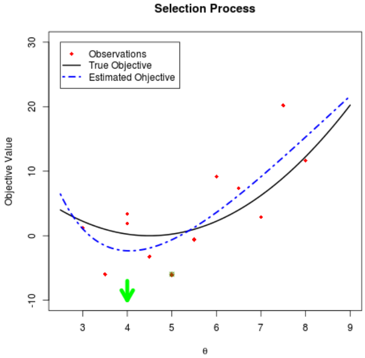


Figure : A Snapshot of the Algorithm

Motivation

Make Use of All Available Information?

Objective Function Approximation

Estimate The Objective Function

If we are dealing with the problem

$$\max_{\theta \in \Theta} J(\theta)$$

$$J(\theta) = E[T(\theta)]$$

$$\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$$

With Observations on Each Alternative

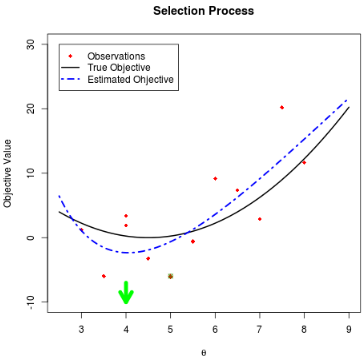


Figure : A Snapshot of the Algorithm

Motivation

Make Use of All Available Information?

If we are dealing with the problem

$$\max_{\theta \in \Theta} J(\theta)$$

$$J(\theta) = E[T(\theta)]$$

$$\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$$

With Observations on Each Alternative

Objective Function Approximation

Estimate The Objective Function

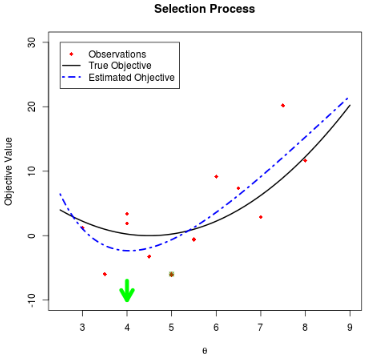


Figure : A Snapshot of the Algorithm

Outline

Design of DVA algorithm

How will function estimation help us?

We want to achieve three major things
1)Assign Resources to Potentially Good Alternatives
2) Assign Resources to Estimate the Objective Function More Accurately
3) Assign Resources to Eliminate False Good ones

And we choose our allocation rules to be
1) we simulate on the estimated optimal alternative
2) simulate on the alternative with current optimal mean
3) simulate on the alternative with maximum bias from the estimated objective

Design of DVA algorithm

How will function estimation help us?

We want to achieve three major things
1)Assign Resources to Potentially Good Alternatives
2) Assign Resources to Estimate the Objective Function More Accurately
3) Assign Resources to Eliminate False Good ones

And we choose our allocation rules to be
1) we simulate on the estimated optimal alternative
2) simulate on the alternative with current optimal mean
3) simulate on the alternative with maximum bias from the estimated objective

Design of DVA algorithm

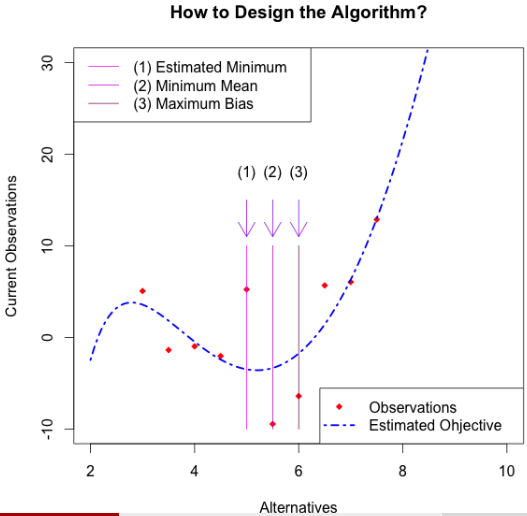
How will function estimation help us?

We want to achieve three major things
1)Assign Resources to Potentially Good Alternatives
2) Assign Resources to Estimate the Objective Function More Accurately
3) Assign Resources to Eliminate False Good ones

And we choose our allocation rules to be
1) we simulate on the estimated optimal alternative
2) simulate on the alternative with current optimal mean
3) simulate on the alternative with maximum bias from the estimated objective

Selection of the Algorithm

Assign Resources According to Current Results



Statement of the Algorithm

a) Simulate one observation on each alternative

b) while we have more resources

1. *Assign resoures according to the above rule*
2. *Update the estimated function*
3. *return to b)*

c) return the estimated optimal

Technical Details in Function Estimation

Recursive Least Square Method

Estimate with least square approach, where the basis functions are designed as

$$\hat{J}(\theta) = a\theta^2 + b\theta + c + \frac{d}{\theta}$$

Least Square is

$$\hat{\beta} = (\Phi^T \Phi)\Phi y$$

Recursive Updates on Matrix are done through

$$\beta^{(N+1)} = \beta^N + \frac{y_{N+1} - \Phi_{N+1}\beta^N}{1 + (\Phi^{N+1})^T B^N (\Phi^{N+1})} B^N$$

$$B^{N+1} = B^N - \frac{B^N \Phi^{N+1} (\Phi^{N+1})^T (B^N)^T}{1 + (\Phi^{N+1})^T B^N (\Phi^{N+1})}$$

Please refer to the project report (initial version done) for the mathematical details and assumptions

Outline

Design an Experiment

similar to the one in the original OCBA paper

$T(\theta) = (\theta - 4.5)^2 + 5Z$

$Z \sim N(0, 1)$

$J(\theta) = (\theta - 4.5)^2$

$Optimal:$

$J(4.5) = 0$

$sub-optimal:$

$J(4) = J(5) = 0.25$

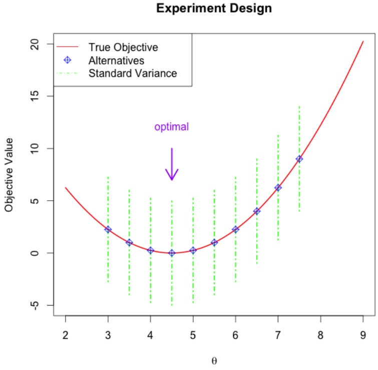


Figure : The Designed Experiment

Design an Experiment

similar to the one in the original OCBA paper

$$T(\theta) = (\theta - 4.5)^2 + 5Z$$
$$Z \sim N(0, 1)$$
$$J(\theta) = (\theta - 4.5)^2$$
$$Optimal:$$
$$J(4.5) = 0$$
$$sub - optimal:$$
$$J(4) = J(5) = 0.25$$

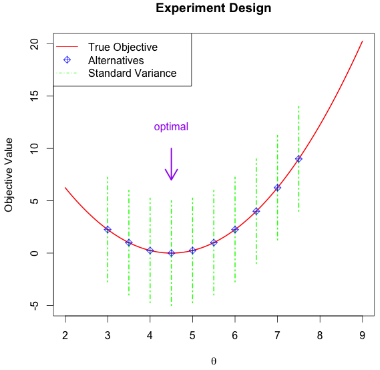


Figure : The Designed Experiment

Design an Experiment

similar to the one in the original OCBA paper

$$T(\theta) = (\theta - 4.5)^2 + 5Z$$
$$Z \sim N(0, 1)$$
$$J(\theta) = (\theta - 4.5)^2$$
$$Optimal:$$
$$J(4.5) = 0$$
$$sub-optimal:$$
$$J(4) = J(5) = 0.25$$

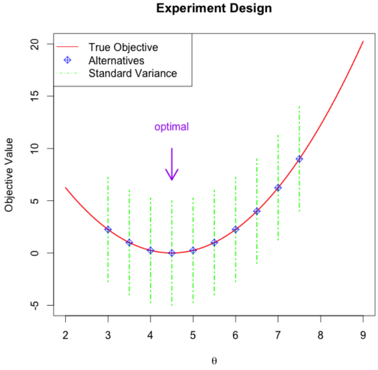
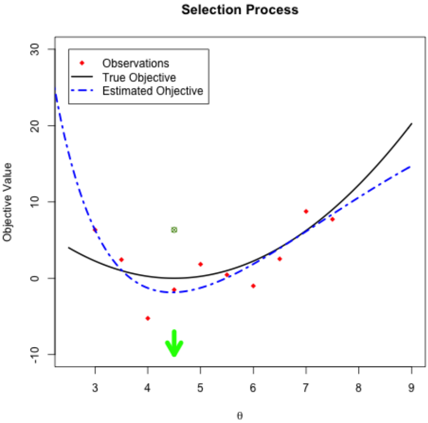


Figure : The Designed Experiment

Animation of the Algorithm

Here We Present an Animation of the Algorithm



Comparison with OCBA and Equal Allocation

DVA outperforms OCBA and EQ

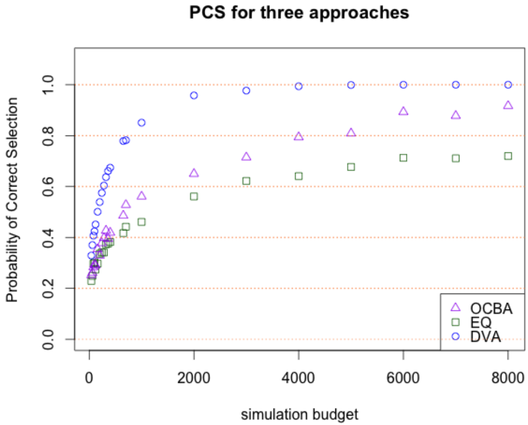


Figure : PCS plot, with 1000 replications under each budget

Possible Problem with the Designed Experiment

many details to consider

Though our method works very well under the designed experiment, there are a couple things to notice

- the true objective function can be estimated unbiasedly from our basis function design

$$\begin{aligned}\hat{J}(\theta) &= a\theta^2 + b\theta + c + \frac{d}{\theta} \\ &= \theta^2 - 9\theta + 20.25\end{aligned}$$

- alternatives has equal variance, which is desirable for least square estimation

But still, we can try to apply this algorithm on realistic problems

Possible Problem with the Designed Experiment

many details to consider

Though our method works very well under the designed experiment, there are a couple things to notice

- the true objective function can be estimated unbiasedly from our basis function design

$$\begin{aligned}\hat{J}(\theta) &= a\theta^2 + b\theta + c + \frac{d}{\theta} \\ &= \theta^2 - 9\theta + 20.25\end{aligned}$$

- alternatives has equal variance, which is desirable for least square estimation

But still, we can try to apply this algorithm on realistic problems

Outline

Outline

Design of the Problem

make use of our course products

M/M/C queue problem

$J(I) = pE(N(I)) - cI$

I : # servers

N : # served customer

$p = 1$, profit from one customer

$c = 4$, cost of training a server

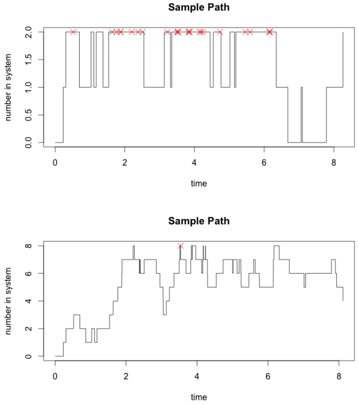
$\lambda = 5$, arrival rate

$\mu = 1$, service rate

J : expected profit in 8 hours

less servers means less customer

More servers, more training cost



Design of the Problem

make use of our course products

M/M/C queue problem

$J(l) = pE(N(l)) - cl$

l : *# servers*

N : *# served customer*

$p = 1$, *profit from one customer*

$c = 4$, *cost of training a server*

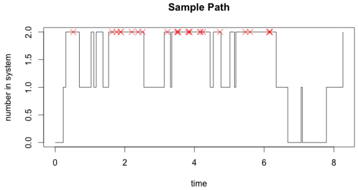
$\lambda = 5$, *arrival rate*

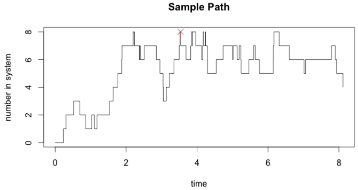
$\mu = 1$, *service rate*

J : *expected profit in 8 hours*

less servers means less customer

More servers, more training cost





Design of the Problem

make use of our course products

M/M/C queue problem

$J(l) = pE(N(l)) - cl$

l : *# servers*

N : *# served customer*

$p = 1$, *profit from one customer*

$c = 4$, *cost of training a server*

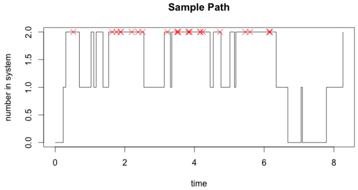
$\lambda = 5$, *arrival rate*

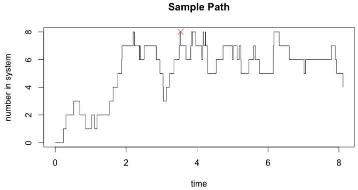
$\mu = 1$, *service rate*

J : *expected profit in 8 hours*

less servers means less customer

More servers, more training cost

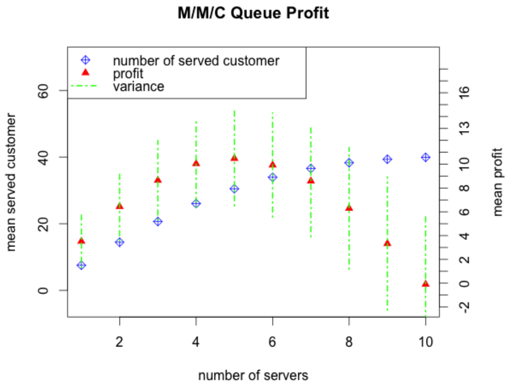




Benchmark of the Problem

let's have a look at the true performance

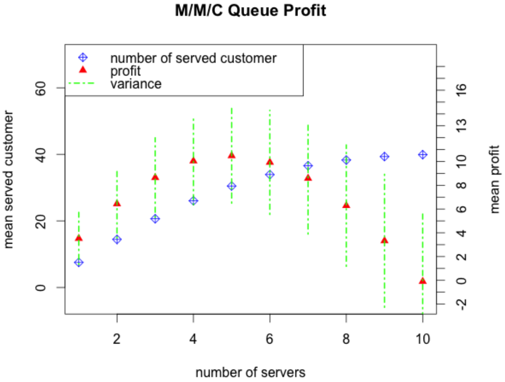
- Choose between alternatives 1 to 10
- Simulate for 1000 replications for each capacity
- Optimal at $l = 5$
- Variance increases



Benchmark of the Problem

let's have a look at the true performance

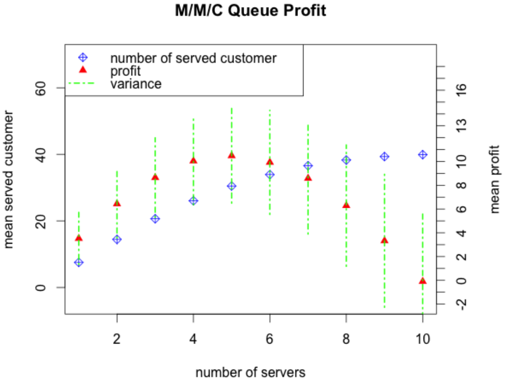
- Choose between alternatives 1 to 10
- Simulate for 1000 replications for each capacity
- Optimal at $l = 5$
- Variance increases



Benchmark of the Problem

let's have a look at the true performance

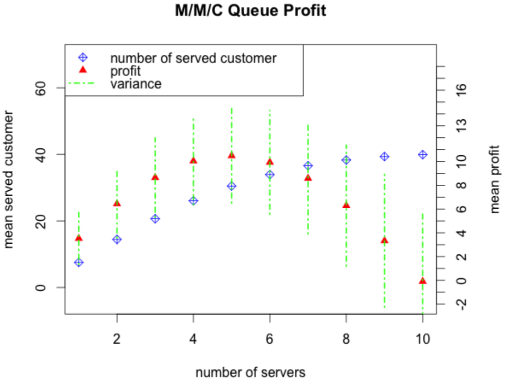
- Choose between alternatives 1 to 10
- Simulate for 1000 replications for each capacity
- Optimal at $l = 5$
- Variance increases



Benchmark of the Problem

let's have a look at the true performance

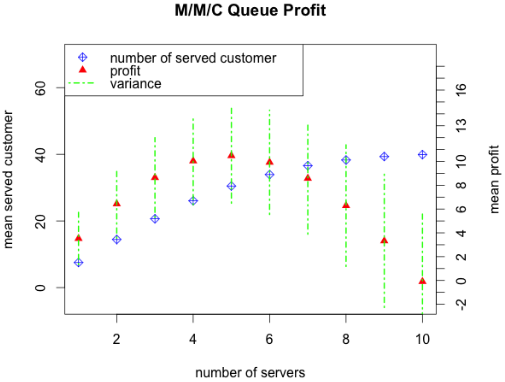
- Choose between alternatives 1 to 10
- Simulate for 1000 replications for each capacity
- Optimal at $l = 5$
- Variance increases



Benchmark of the Problem

let's have a look at the true performance

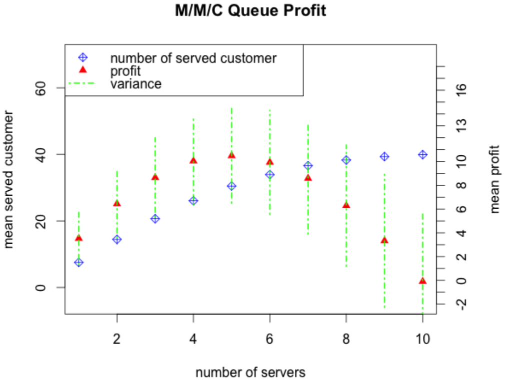
- Choose between alternatives 1 to 10
- Simulate for 1000 replications for each capacity
- Optimal at $l = 5$
- Variance increases



Benchmark of the Problem

let's have a look at the true performance

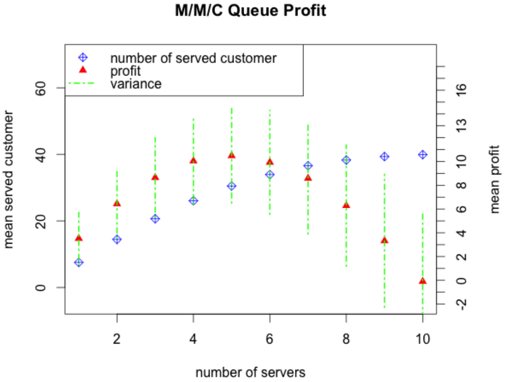
- Choose between alternatives 1 to 10
- Simulate for 1000 replications for each capacity
- Optimal at $l = 5$
- Variance increases



Benchmark of the Problem

let's have a look at the true performance

- Choose between alternatives 1 to 10
- Simulate for 1000 replications for each capacity
- Optimal at $l = 5$
- Variance increases



Performance of DVA Algorithm

DVA method dominates OCBA and EQ

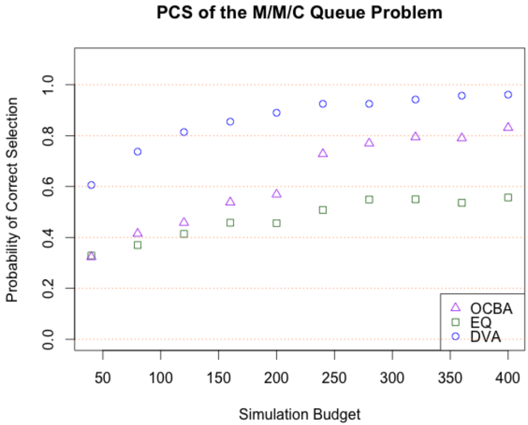


Figure : PCS plot, with 1000 replications under each simulation budget

Outline

Design of the Inventory Problem

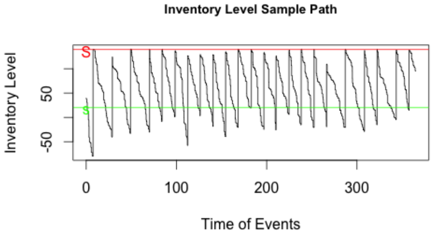
Notation	Specification	Value
d_i	inter demand time	$U[0,1]$
L_i	order lead time	$U[0,1]$
D_i	Demand	$Exp(\frac{1}{5})$
h	weekly holding cost	0.10
p	weekly storage cost	10
c	order set up cost	10
m	ordering cost	0.10

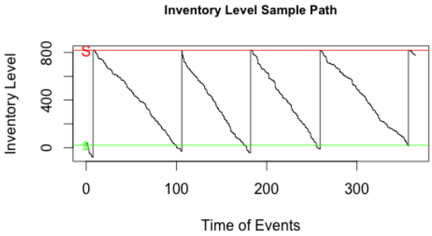
$$Z = h \int I(t)^+ dt - p \int I(t)^- dt + c \sum_{reviews} I_{\{X_i > 0\}} + m \sum_{reviews} X_i$$

Specification of the Problem

We are interested in
1) Minimum Yearly cost
2) Setting s=20
3) Choose
$S \in \{80, 140, 200, 300, 400$
$500, 600, 700, 820, 950\}$

- S too big means too much holding cost
- S too small means too much ordering cost

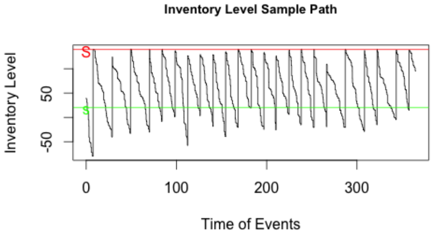


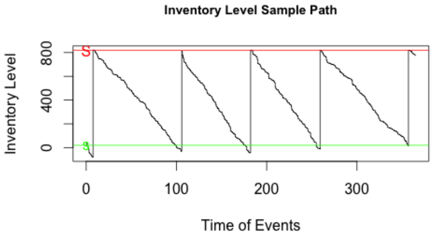


Specification of the Problem

We are interested in
1) Minimum Yearly cost
2) Setting s=20
3) Choose
$S \in \{80, 140, 200, 300, 400$
$500, 600, 700, 820, 950\}$

- S too big means too much holding cost
- S too small means too much ordering cost

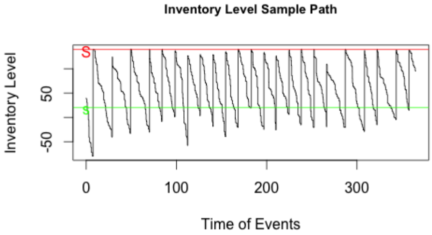


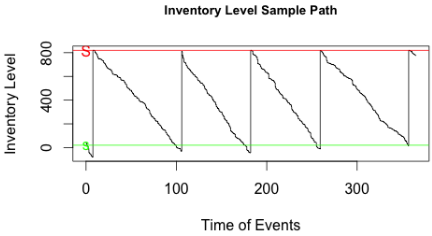


Specification of the Problem

We are interested in
1) Minimum Yearly cost
2) Setting s=20
3) Choose
$S \in \{80, 140, 200, 300, 400$
$500, 600, 700, 820, 950\}$

- S too big means too much holding cost
- S too small means too much ordering cost

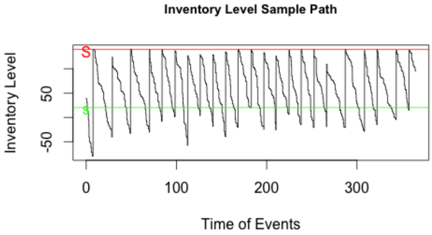


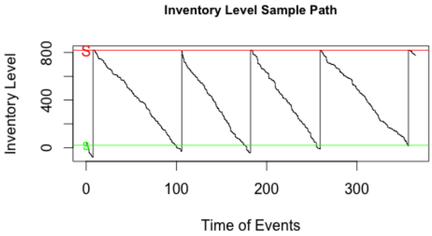


Specification of the Problem

We are interested in
1) Minimum Yearly cost
2) Setting s=20
3) Choose
$S \in \{80, 140, 200, 300, 400$
$500, 600, 700, 820, 950\}$

- S too big means too much holding cost
- S too small means too much ordering cost

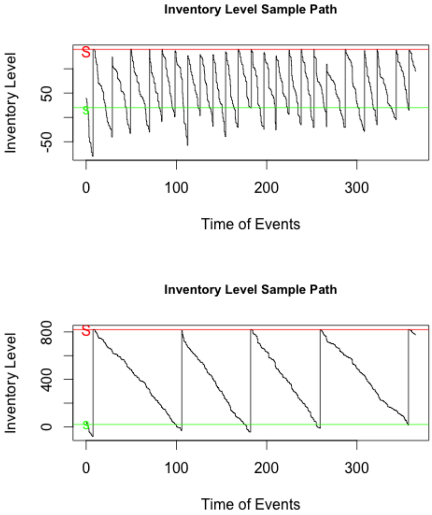




Specification of the Problem

We are interested in
1) Minimum Yearly cost
2) Setting s=20
3) Choose
$S \in \{80, 140, 200, 300, 400$
$500, 600, 700, 820, 950\}$

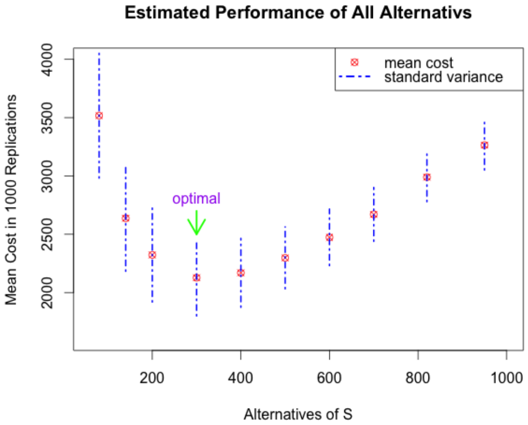
- S too big means too much holding cost
- S too small means too much ordering cost



Benchmark for the Problem

the correct information about this problem

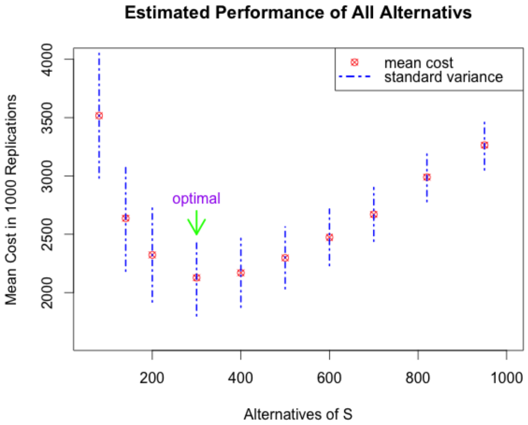
1. Simulate 1000 replications for each alternative
2. Variance decreases
3. Optimal at S=300
4. True objective is not a parabola



Benchmark for the Problem

the correct information about this problem

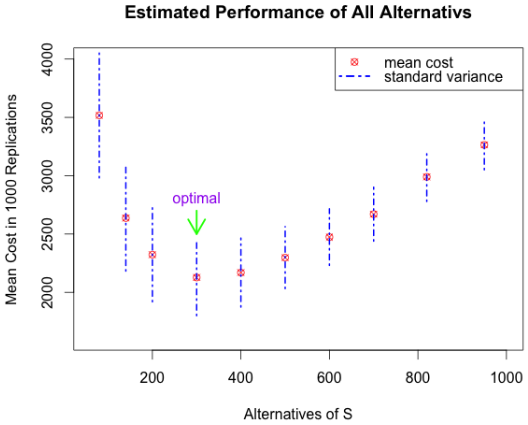
1. Simulate 1000 replications for each alternative
2. Variance decreases
3. Optimal at S=300
4. True objective is not a parabola



Benchmark for the Problem

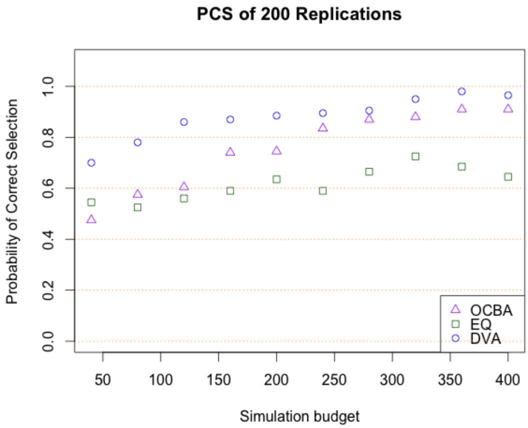
the correct information about this problem

1. Simulate 1000 replications for each alternative
2. Variance decreases
3. Optimal at S=300
4. True objective is not a parabola



Performance of the Algorithm

DVA outperforms OCBA and EQ



Outline

What We Have Done

- We designed the DVA algorithm based on least square function approximation
- We applied our algorithm to a designed experiment, a M/M/C queueing system and a (s,S) inventory system
- We showed through simulation that our algorithm performs better than OCBA and Equal Allocation in those problems
- This is expected because DVA algorithm considers more information when doing resources allocation.

What We Have Done

- We designed the DVA algorithm based on least square function approximation
- We applied our algorithm to a designed experiment, a M/M/C queueing system and a (s,S) inventory system
- We showed through simulation that our algorithm performs better than OCBA and Equal Allocation in those problems
- This is expected because DVA algorithm considers more information when doing resources allocation.

What We Have Done

- We designed the DVA algorithm based on least square function approximation
- We applied our algorithm to a designed experiment, a M/M/C queueing system and a (s,S) inventory system
- We showed through simulation that our algorithm performs better than OCBA and Equal Allocation in those problems
- This is expected because DVA algorithm considers more information when doing resources allocation.

What We Have Done

- We designed the DVA algorithm based on least square function approximation
- We applied our algorithm to a designed experiment, a M/M/C queueing system and a (s,S) inventory system
- We showed through simulation that our algorithm performs better than OCBA and Equal Allocation in those problems
- This is expected because DVA algorithm considers more information when doing resources allocation.

What We Can Do Next

There are indeed a lot of things to do

- the selection rule is subjective, you can have other designs
- how to consider irregular points? i.e. what if the true objective function is highly irregular
- design of the return. You can either return the estimated optimal or the observed optimal. In the later case, DVA is only a way of resource allocation.
- how to design a way of constructing a set of coordinates for more complex ranking and selection problems so that the true objective function is kind of regular?

What We Can Do Next

There are indeed a lot of things to do

- the selection rule is subjective, you can have other designs
- how to consider irregular points? i.e. what if the true objective function is highly irregular
- design of the return. You can either return the estimated optimal or the observed optimal. In the later case, DVA is only a way of resource allocation.
- how to design a way of constructing a set of coordinates for more complex ranking and selection problems so that the true objective function is kind of regular?

What We Can Do Next

There are indeed a lot of things to do

- the selection rule is subjective, you can have other designs
- how to consider irregular points? i.e. what if the true objective function is highly irregular
- design of the return. You can either return the estimated optimal or the observed optimal. In the later case, DVA is only a way of resource allocation.
- how to design a way of constructing a set of coordinates for more complex ranking and selection problems so that the true objective function is kind of regular?

What We Can Do Next

There are indeed a lot of things to do

- the selection rule is subjective, you can have other designs
- how to consider irregular points? i.e. what if the true objective function is highly irregular
- design of the return. You can either return the estimated optimal or the observed optimal. In the later case, DVA is only a way of resource allocation.
- how to design a way of constructing a set of coordinates for more complex ranking and selection problems so that the true objective function is kind of regular?

Workload

this is an interesting problem

- 19 frequently used R scripts
- about 50 frequently used R functions
- 7 days of coding, debugging and simulation
- we also experimented on a stepsize sequence we derived for SA, but the result is not so good.

Workload

this is an interesting problem

- 19 frequently used R scripts
- about 50 frequently used R functions
- 7 days of coding, debugging and simulation
- we also experimented on a stepsize sequence we derived for SA, but the result is not so good.

Workload

this is an interesting problem

- 19 frequently used R scripts
- about 50 frequently used R functions
- 7 days of coding, debugging and simulation
- we also experimented on a stepsize sequence we derived for SA, but the result is not so good.

Workload

this is an interesting problem

- 19 frequently used R scripts
- about 50 frequently used R functions
- 7 days of coding, debugging and simulation
- we also experimented on a stepsize sequence we derived for SA, but the result is not so good.

End of Presentation
Thank you