Stochastic mixed discrete and continuous optimization for constrained problems.

Larry Fenn and Felisa Vázquez-Abad
Computer Science, Hunter College and GC

SSMO CUNY

CUNY Institute for
Computer Simulation, Stochastic
Modeling and Optimization

INFORMS Annual Meeting, Philadelphia, Nov 4, 2015

Problem Formulation

Resource: integer variable $b \in \mathbb{N}$.

Operating control: continuous variable $u \in \mathbb{R}^d$.

Constraint: $\mathcal{G}(b, u) \leqslant c$.

Operating costs increase with resources.

Assumption

For each b *the constraint function* $\mathcal{G}(b, \cdot) \in C^2$ *and has a unique minimum. Also,* $f(b)$ *is monotone decreasing in* b*, where*

$$f(b) = \min_u \mathcal{G}(b, u).$$

Problem Formulation

Resource: integer variable $b \in \mathbb{N}$.

Operating control: continuous variable $u \in \mathbb{R}^d$.

Constraint: $\mathcal{G}(b, u) \leqslant c$.

Operating costs increase with resources.

Assumption

For each b *the constraint function* $\mathcal{G}(b, \cdot) \in C^2$ *and has a unique minimum. Also,* $f(b)$ *is monotone decreasing in* b*, where*

$$f(b) = \min_u \mathcal{G}(b, u).$$

Simplified formulation: Seek the solution to the problem:

$$\min_b f(b) \qquad \text{subject to } f(b) \leqslant c$$

Problem Formulation

Resource: integer variable $b \in \mathbb{N}$.
Operating control: continuous variable $u \in \mathbb{R}^d$.
Constraint: $\mathcal{G}(b, u) \leqslant c$.
Operating costs increase with resources.

Assumption

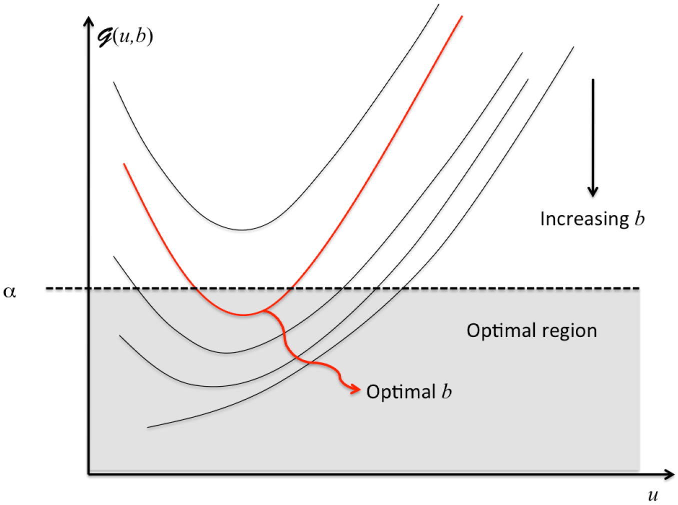
For each b *the constraint function* $\mathcal{G}(b, \cdot) \in C^2$ *and has a unique minimum. Also,* $f(b)$ *is monotone decreasing in* b*, where*

$$f(b) = \min_u \mathcal{G}(b, u).$$

Simplified formulation: Seek the solution to the problem:

$$\min_b f(b) \qquad \text{subject to } f(b) \leqslant c$$

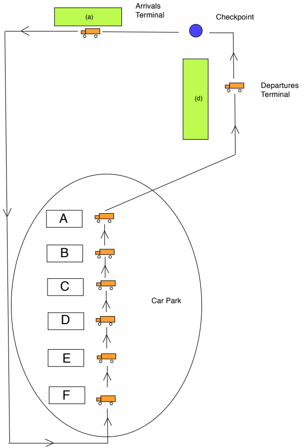
But... we do not have direct measurements or closed form expression of $\mathcal{G}$. It depends on an underlying stochastic process.



Probability constraint for fixed fleet size

Motivation: example

- Melbourne airport, 2005.
- *Fleet size* b to be determined
- *Headway control:* buses are at least u minutes apart
- Arrivals: passengers destined to retrieve their cars
- Passengers queue at carpark stations (destination: Departures)
- Buses loop around carpark stations
- Unload first, load next
- Random loading/unloading times
- Buses empty at Departures



Motivation: example

Quality of Service criterion: the 95/10 rule

At least 95% of the passengers wait less than 10 minutes for a bus.

- Implicit optimal scheduling of buses (headway control u).

Motivation: example

Quality of Service criterion: the 95/10 rule

At least 95% of the passengers wait less than 10 minutes for a bus.

- Implicit optimal scheduling of buses (headway control u).
- If b buses can satisfy the constraint, then so do b' buses for all $b' \geqslant b$ (monotone constraint)

Motivation: example

Quality of Service criterion: the 95/10 rule

At least 95% of the passengers wait less than 10 minutes for a bus.

- Implicit optimal scheduling of buses (headway control u).
- If b buses can satisfy the constraint, then so do b' buses for all $b' \geqslant b$ (monotone constraint)
- Constraint:

$$\mathcal{G}(b, u) = \lim_{T \to \infty} \mathbb{E} \left(\frac{1}{\lambda T} \sum_{i=1}^{N(T)} \mathbf{1}_{\{W_i > 10\}} \right),$$

λ arrivals per unit time, $N(\cdot)$ arrival process. W_i is the waiting time of passenger i at his/her station queue.

Model Assumptions

Given resource/control parameters (b, u), the *underlying process* $\{\xi_n\}$ is a Markov chain on $\mathcal{S} \subset \mathbb{R}^n$.

Transition probabilities:

$$p_{b,u}(x; dx) = \mathbb{P}(\xi_{n+1} \in dx \,|\, \xi_n = x)$$

Assume stationary measure $\mu_{b,u}$, and constraint of the form

$$\mathcal{G}(b, u) = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} g(\xi_n) = \mathbb{E}_{\mu_{b,u}}(g(\xi_n)).$$

Assumption. CLT on g in order to obtain steady-state mean and asymptotic variance.

In practice: use long simulations to approximate $\mathcal{G}(b, u)$.

Formulation

We present a simpler problem where $u \geqslant 0$ is one-dimensional.

$$f(b) = \min_{u \in \mathbb{R}^+} \mathcal{G}(b, u)$$
$$b^* = \arg\min(f(b) : \ f(b) \leqslant c)$$

Constraint $\mathcal{G}(b, u)$ is estimated (long simulations).

Objective: Find an *efficient* method for fast approximation with

- A tolerance level ϵ for the estimation of $\widehat{f}(b^*)$.
- A statistical confidence level α for $\mathrm{PCS}(b^*)$.

Main Contributions

- Comparison of various methods to approximate $\mathtt{b}^*$.
- *Intellectual merit.* We analyze convergence for:
  1. PCS for $\mathtt{b}^*$,
  2. Computational complexity (number of iterations),
  3. Estimate of final error in $\mathtt{f}(\mathtt{b}^*)$.
- *Potential impact.* The problem arises naturally in large systems where the number of resources may be very large (public transportation, number of servers in large networks, personnel allocation for health management, etc).

Simplified Formulation

Observation. For fixed b, the model is $f(b) = \min_u \mathcal{G}(b, u)$.

- We are interested in finding b^*: smallest resource allocation that satisfies constraint.
- When $b > b^*$ it suffices to find u such that $\mathcal{G}(b, u) < c$ to determine that b satisfies constraint.
- Therefore, there is *no need* to find exact minimum.

Binary Search

The *outer loop* is a binary problem:

Does b_n satisfy constraint?

yes: choose $b_{n+1} < b_n$

no: choose $b_{n+1} > b_n$

Adapt *binary search* algorithm on b to stochastic case.

Method 1: target tracking

Assume that it is always possible to *overestimate* $u^*(b)$. Given b_n and $u_n(0)$ "large", use target tracking:

$$u_n(k+1) = u_n(k) - \eta(\widehat{\mathcal{G}}(b_n, u_n(k)) - c); \quad \eta > 0$$

Behavior:

- If $b > b^*$ $u_n(k)$ will decrease towards constraint satisfaction.
- If $b < b^*$ $u_n(k) \to 0$ and $\mathcal{G}(b_n, u_n(k)) > c$ always.

Open problem: how to choose algorithm parameter: step size η.

Method 2: truncated golden search

Assume initial interval $[\ell(0), r(0)] = [0, \bar{u}]$ for all b.
$\varphi = (\sqrt{5} - 1)/2$. Given b_n and tolerance κ, initialize

$$x(0) = r(0) - \varphi(\ell(0) - r(0))$$
$$y(0) = \ell(0) + \varphi(r(0) - \ell(0))$$

If $\mathcal{G}(b_n, x(k)) < \mathcal{G}(b_n, y(k))$ then *erase right subinterval*:
$r(k+1) = y(k); y(k+1) = x(k); x(k+1) = r(k+1) - \varphi(\ell(k+1) - r(k+1))$.

Otherwise *erase left subinterval*:
$\ell(k+1) = x(k); x(k+1) = y(k); y(k+1) = \ell(k+1) + \varphi(\ell(k+1) - r(k+1))$.

Stop when either constraint is satisfied or when $r(k) - \ell(k) \leqslant \kappa$.

Method 3: gradient search

Assume that we can estimate $\mathcal{G}'(b,u)$ (can be FD's). Given b_n and $u_n(0)$ iterate with:

$$u_n(k+1) = u_n(k) - \eta\, \widehat{\mathcal{G}}'(b_n, u_n(k)); \quad \eta > 0$$

Then under some technical assumptions $\lim_k u_n(k)$ approaches $u^*(b)$ (in some adequate topology).

Open problem: How to choose algorithm parameter step size η and stopping criterion?

- Method 1: Target tracking
- Method 2: Golden section search
- Method 3: Gradient search

Deterministic Golden Section Search

Theorem

Let $g_b(x) = \mathcal{G}(b, x)$, $x \in [0, m]$.
Assume that g_b *is twice-differentiable, with a constant* K *such that* $0 < g_b''(x) < \frac{1}{K}$. *Golden section will achieve* ϵ *tolerance* $(K \geqslant \epsilon > 0)$ *at iteration*

$$n(K, \epsilon) \geqslant \frac{\log(K^2 - (K - \epsilon)^2) - \log m}{\log \varphi}$$

Stochastic Golden Section Search

Theorem

Let $f(b) = \min_x g_b(x)$. *Suppose only noisy observations* $\hat{g}_b(x)$ *are available for* $g_b(x)$. *Assume the following are true for* g_b *and* $\hat{g}_b$:

- g_b *is twice-differentiable, with a constant* K *such that* $0 < g_b''(x) < \frac{1}{K}$.
- $\hat{g}_b(x) = g_b(x) + Z_{b,x}$ *where each* $Z_{b,x}$ *is independent and normally distributed,* $\sigma^2 \geqslant \mathbb{V}\mathrm{ar}(Z_{b,x})$.

If the number of samples $n \geqslant (n(K,\epsilon)+1)\left(\frac{c\sigma}{\epsilon}\right)^2$

then $\mathbb{P}(|\hat{f}(b) - f(b)| \geqslant \epsilon) \leqslant \alpha$

where $c = \Phi^{-1}\left(\frac{1}{2}\left(1 + \exp\left(\frac{\log(1-\alpha)}{n(K,\epsilon)+1}\right)\right)\right)$

Stochastic Binary Search

Suppose that a strictly decreasing real-valued function f is defined on the discrete domain $[1, 2, ..., N]$ and that there is a procedure for observations of $\hat{f}$ to be generated with the following property for any arbitrary value C and for any $\epsilon > 0$ and $0 < \alpha \leqslant 1$:

$$\mathbb{P}\left(f(i) > C \,|\, \hat{f}(i) < C - \epsilon\right) \leqslant \alpha \text{ (False positive)}$$

$$\mathbb{P}\left(f(i) < C \,|\, \hat{f}(i) > C + \epsilon\right) \leqslant \alpha \text{ (False negative)}$$

Theorem

For binary search to succeed with probability $1 - \beta$, $\hat{f}$ must be determined such that the α in the above conditions satisfies

$$\alpha \leqslant 1 - \left(\exp\left(\frac{\log(1-\beta)}{\log_2\lceil N\rceil}\right)\right)$$

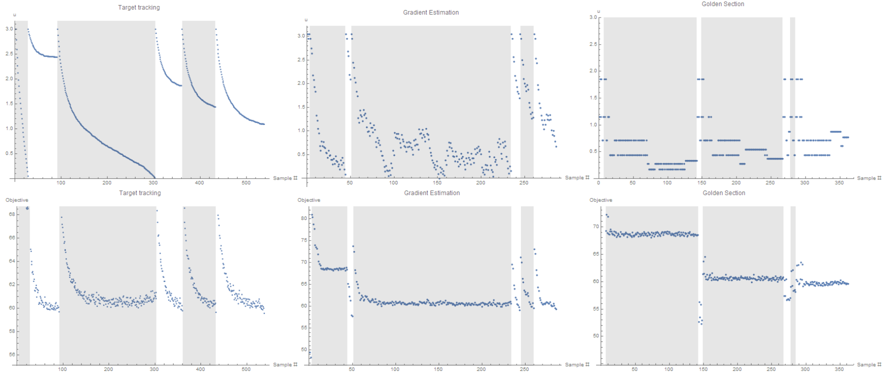
Error Estimates

Theorem

With the same assumptions as before, an upper bound for expected error (in the search domain) for stochastic golden search with probability of correct selection $1-\alpha$*, and* $n(K,\epsilon), m, \varphi$ *as before is:*

$$\varphi(1-\varphi^{n})m\left(1-\exp\left(\frac{\log(1-\alpha)}{n(K,\epsilon)+1}\right)\right)$$

Comparison of Algorithms

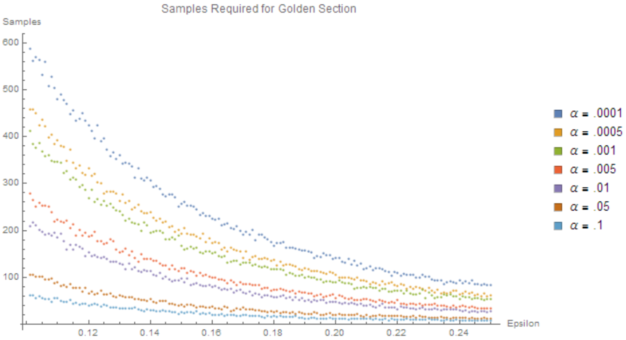


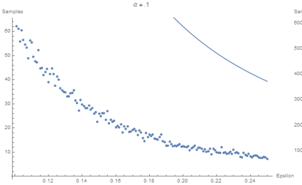
Experimental Results

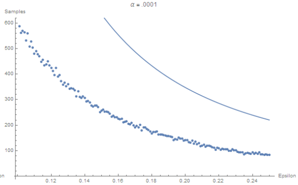
Comparison of methods: average performance over 10,000 trials

Method	CPU	Samples	PCS	MSE
RM	183.2	593.8	.7661	.6624
KW	177.2	604.4	.7113	.2967
GS	729.7	420.2	.9960	.2729

Samples: Observed and Bounds







Concluding Remarks

On-going and future extensions.

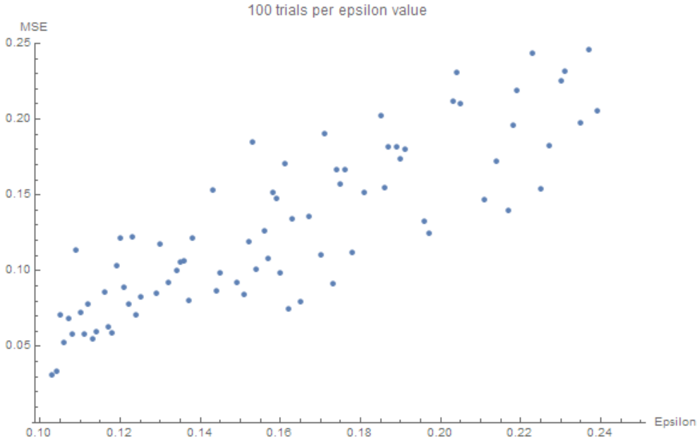
- Full comparison of methods: learn dependence on problem characteristics.
- Generalization to multidimensional $u \in \mathbb{R}^d$: Golden search on random directions.
- Error detection and backtracking.
- Parallel computation for accelerated golden section search and backtracking implementation.

Thank You for your Attention

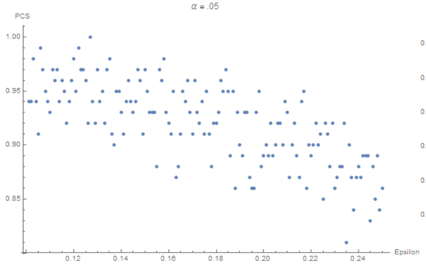
Questions?

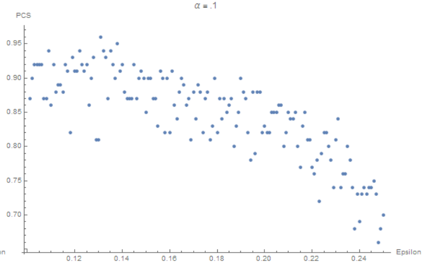
CUNY Institute for
Computer Simulation, Stochastic
Modeling and Optimization

Epsilon vs. MSE



Alpha vs. PCS


$\alpha = .05$
PCS
1.00
0.95
0.90
0.85
0.12
0.14
0.16
0.18
0.20
0.22
0.24
Epsilon


$\alpha = .1$
PCS
0.95
0.90
0.85
0.80
0.75
0.70
0.12
0.14
0.16
0.18
0.20
0.22
0.24
Epsilon