Asynchronous Knowledge Gradient Policy for Ranking and Selection

Bogumił Kamiński

Decision Analysis and Support Unit
Warsaw School of Economics

October 25, 2017

Ranking & selection problem

1. We want to choose from N alternatives
2. Every alternative has some true and unknown value $\tilde{\mu}_i$
3. The objective is to find the alternative that has the highest value
4. Measurement of each alternative is associated with IID noise following $N(0, \sigma_\varepsilon^2)$

Applications

Various flavors of A/B testing, examples:

1. website design
2. clinical trials
3. *discrete simulation optimization*


SIMCITY QUICK LINKS
REGISTER · LOGIN · HELP
SIMCITY
AVAILABLE NOW!
NEWSLETTER SIGNUP
GAME INFO
NEWS
COMMUNITY
MEDIA
BUY NOW
PRE-ORDER
AND GET $20 OFF
YOUR NEXT PURCHASE*
SIMCITY™
$59.99
PC Download
PC Physical
BUY NOW
SIMCITY™ DIGITAL DELUXE EDITION
$79.99
PC Download
BUY NOW
DIGITAL DELUXE EDITION
INCLUDES
HEROES AND VILLAINS SET
FRENCH CITY SET
GERMAN CITY SET
BRITISH CITY SET

Earlier work

What does "selecting best alternative" mean?

1. maximize probability of correct selection: $\Pr(\tilde{\mu}_s = \max_i \tilde{\mu}_i)$
2. minimize expected loss: $E(\max_i \tilde{\mu}_i - \tilde{\mu}_s)$

if s is a selected alternative.

Standard algorithms:

1. single step rules (e.g. allocate all budget equally to all alternatives)
2. two-step rules
3. batch rules (e.g. optimal computing budget allocation)
4. *one-step ahead sequential rules (e.g. Knowledge Gradient)*

Knowledge Gradient policy (1)

1. Bayesian approach
2. Prior beliefs

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix} = N\left(\begin{bmatrix} \mu_{(0),1} \\ \mu_{(0),2} \\ \vdots \\ \mu_{(0),N} \end{bmatrix}, \begin{bmatrix} \sigma^2_{(0),1} & 0 & \cdots & 0 \\ 0 & \sigma^2_{(0),2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{(0),N} \end{bmatrix} \right)$$

3. Sequential policy
4. One measurement at a time policy

Knowledge Gradient policy (2)

1. At step k we measure alternative i and obtain value y we update our beliefs

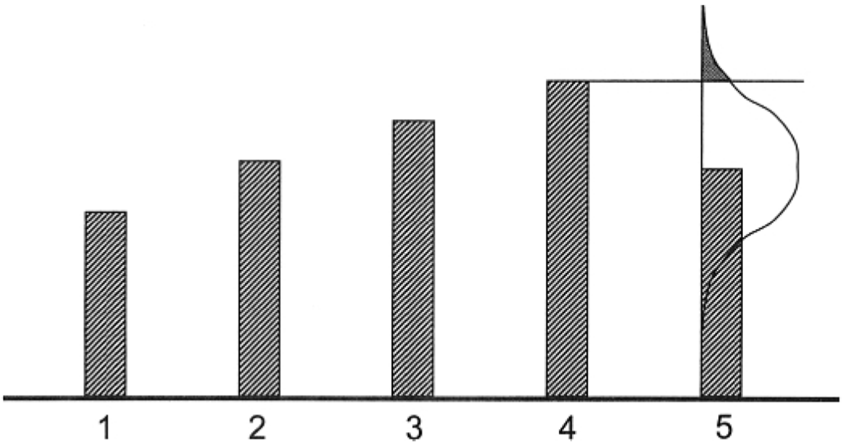
$$\sigma^2_{(k),i} = \left(\frac{1}{\sigma^2_{(k-1),i}} + \frac{1}{\sigma^2_\varepsilon} \right)^{-1}$$

$$\mu_{(k),i} = \left(\frac{\mu_i^{(k-1)}}{\sigma^2_{(k-1),i}} + \frac{y}{\sigma^2_\varepsilon} \right) \sigma^2_{(k),i}$$

2. If we decide stop after k steps we choose $\arg\max_i \mu_{(k),i}$
3. How to select which alternative to measure at step k?

$$\arg\max_i E(\max_j \mu_{(k),j})$$

Knowledge Gradient policy: idea



Asynchronous parallelization

Idea

Extend the Knowledge Gradient policy to allow for parallel evaluation of alternatives.
Possible approaches:

synchronous new tasks are assigned to all workers at the same time

asynchronous every worker is independently assigned a new task

Why asynchronous?

- high variance of execution time of individual computation
- heterogeneous computing power across nodes in a cluster
- worker failures need to be handled

What do we expect from the AKG policy?

1. the expected improvement of quality of the solution per measurement is lower in AKG than in KG
2. the expected time to reach the desired quality of solution is lower in AKG than in KG

Asynchronous Knowledge Gradient (1)

- We have a set W denoting a pool of workers
- We want to assign a new task to $\mathit{worker} \in W$ immediately after it becomes available
- When deciding we have to consider jobs already running in parallel
- The number of scheduled but not observed measurements of alternatives is $\mathbf{s} = (s_1, s_2, \ldots, s_N)$
- Choose to measure alternative that provides the highest expected increase of outcome conditional on $\mathbf{s}$

Asynchronous Knowledge Gradient (2)

Decision rule:

$$AKG(k, \mathbf{s}) = \arg\max_{i \in \{1,2,\ldots,N\}} E\left(V_{(k)|\mathbf{s}+\mathbf{e}_i}\right)$$

where:

$$V_{(k)|\mathbf{s}} = \max_{i \in \{1,2,\ldots,N\}} M_{(k),i|s_i}.$$

and $M_{(k),i|s_i}$ is distribution of beliefs about μ_i conditional on the fact that it will be measured s_i times.
It can be shown that:

- if $s_i = 0$ then the distribution is concentrated at $\mu_{(k),i}$
- if $s_i > 0$ then the distribution is normal with mean $\mu_{(k),i}$ and variance:

$$\sigma^2_{(k),i|s_i} = \sigma^2_{(k),i} - \left(\frac{1}{\sigma^2_{(k),i}} + \frac{s_i}{\sigma^2_\varepsilon}\right)^{-1}.$$

Computational approach

Denote CDF of $M_{(k),i|s_i}$ by $F_{(k),i|s_i}(x)$.

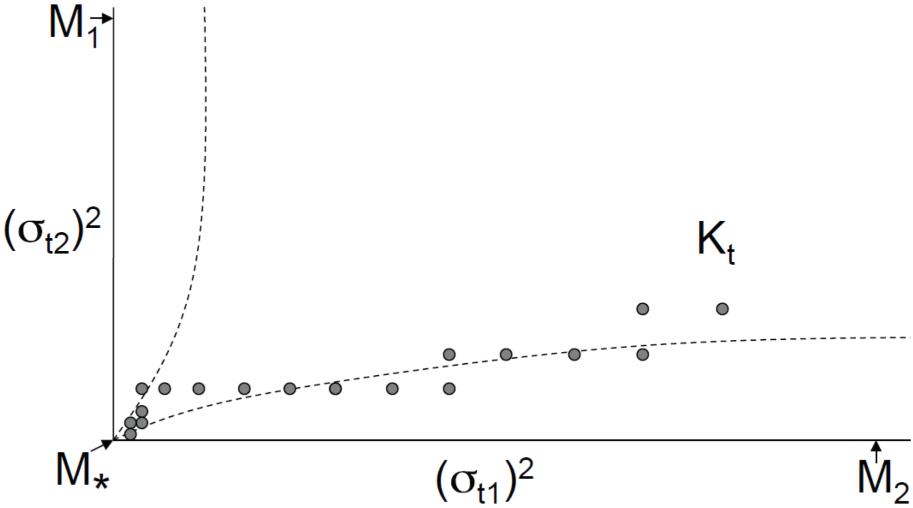
Let:

$$C_{(k)|\mathbf{s}}(x) = \prod_i F_{(k),i|s_i}(x)$$

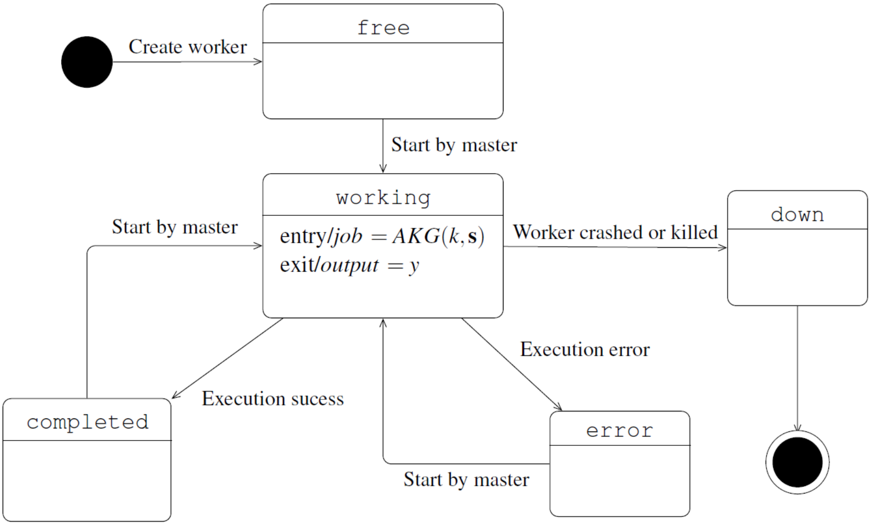
then

$$E(V_{(k)|\mathbf{s}}) = \int_0^1 \frac{1 - C_{(k)|\mathbf{s}}\left(t^{-1} - 1\right) - C_{(k)|\mathbf{s}}\left(1 - t^{-1}\right)}{t^2} dt.$$

AKG is convergent



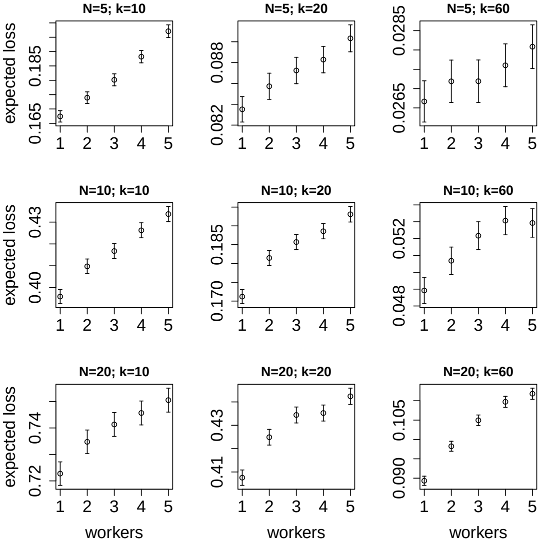
Worker state diagram in master–slave architecture



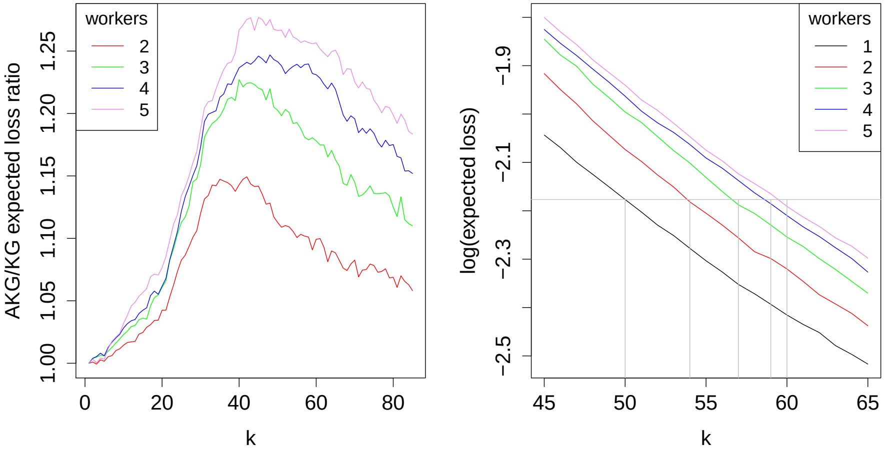
Scenarios for simulation experiments

algorithm	N	workers — $\|W\|$	steps — k	repetitions
KG	$\{5, 10\}$	1	85	114122
AKG	$\{5, 10\}$	$\{1, 2, 3, 4, 5\}$	85	114122
KG	20	1	85	107560
AKG	20	$\{1, 2, 3, 4, 5\}$	85	107560

Expected loss by number of workers $|W|$ (1)



Expected loss by number of workers $|W|$ (2)



Concluding remarks

Main results

- modification of the Knowledge Gradient policy to asynchronous execution
- procedure is numerically traceable and simple to execute in master–slave architecture . . .
- . . . a care is required to cleanup computations
- appealing scaling properties with number of workers

More details

B. Kamiński, P. Szufel: On parallel policies for ranking and selection problems, Journal of Applied Statistics, 2017, doi:10.1080/02664763.2017.1390555