Bayesian Optimization via Simulation with Pairwise Sampling and Correlated Prior Beliefs

Jing Xie
Peter I. Frazier
Stephen E. Chick

Operations Research & Information Engineering, Cornell University
Technology & Operations Management Area, INSEAD

Oct. 8, 2013
2013 INFORMS Annual Meeting, Minneapolis
Discrete Optimization via Simulation (DOvS)

- $A$ is some discrete set.
- Objective function $f : A \mapsto \mathbb{R}$.
- Our goal is to solve

$$\max_{x \in A} f(x).$$
Discrete Optimization via Simulation (DOvS)

- $A$ is some discrete set.
- Objective function $f : A \mapsto \mathbb{R}$.
- Our goal is to solve
  $$\max_{x \in A} f(x).$$

- We cannot evaluate $f(x)$ directly.
- We have a stochastic simulator that can evaluate $f(x)$ with noise.
- It gives us $g(x, \omega) = f(x) + \epsilon(x, \omega)$, where $\mathbb{E}[g(x, \omega)] = f(x)$. 
Example Applications using Discrete-Event Simulations

$A$ may correspond to a $\mathbb{Z}^d$ lattice:
- staffing levels in a hospital
- inventory policies for a supply chain
- admission controls in a call center

$A$ can also correspond to a set of combinatorial structures:
- configurations of an assembly line
We Employ a Bayesian Value-of-Information Approach
We Employ a Bayesian Value-of-Information Approach

- Given the function evaluations obtained, use Bayesian statistics to get:
  - estimates of $f(x)$ over set $A$.
  - uncertainties in these estimates.

A common method: (discrete) Gaussian Process (GP) regression:

- Use these estimates and uncertainties to quantify the contribution of possible future evaluations.
- Decide where to evaluate next.

VOI: Information is valued according to the expected improvement it produces in some decision to be made later. (Raiffa & Schlaifer 1961)
We Employ a Bayesian Value-of-Information Approach

- Given the function evaluations obtained, use Bayesian statistics to get:
  - estimates of $f(x)$ over set $A$.
  - uncertainties in these estimates.
- A common method: (discrete) Gaussian Process (GP) regression:
We Employ a Bayesian Value-of-Information Approach

- Given the function evaluations obtained, use Bayesian statistics to get:
  - estimates of $f(x)$ over set $A$.
  - uncertainties in these estimates.
- A common method: (discrete) Gaussian Process (GP) regression:
  - Use these estimates and uncertainties to
    - quantify the contribution of possible future evaluations.
    - decide where to evaluate next.
We Employ a Bayesian Value-of-Information Approach

- Given the function evaluations obtained, use Bayesian statistics to get:
  - estimates of $f(x)$ over set $A$.
  - uncertainties in these estimates.
- A common method: (discrete) Gaussian Process (GP) regression:

\[ \text{value} \]
\[ x \]

- Use these estimates and uncertainties to
  - quantify the contribution of possible future evaluations.
  - decide where to evaluate next.

**VOI:** Information is valued according to the expected improvement it produces in some decision to be made later. (Raiffa & Schlaifer 1961)
Our Problem Settings

- We consider DOvS over a finite set of alternatives 1, 2, ..., k.
Our Problem Settings

- We consider DOvS over a **finite** set of alternatives $1, 2, \ldots, k$.

- Our simulator allows **correlated** output through common random numbers (CRN).
Our Problem Settings

- We consider DOvS over a finite set of alternatives 1, 2, ..., k.

- Our simulator allows correlated output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector $\theta = (\theta_1, \ldots, \theta_k)$,
  - covariance matrix $\Lambda$ (assumed known, but can be relaxed).

GOAL: find $x^* = \arg\max_x \{\theta^T x\}$.
Our Problem Settings

- We consider DOvS over a finite set of alternatives 1, 2, \ldots, k.

- Our simulator allows correlated output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector \( \theta = (\theta_1, \ldots, \theta_k) \),
  - covariance matrix \( \Lambda \) (assumed known, but can be relaxed).

- GOAL: find \( x^* = \arg\max_x \{ \theta_x \} \).
Our Problem Settings

- We consider DOvS over a **finite** set of alternatives $1, 2, \ldots, k$.

- Our simulator allows **correlated** output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector $\theta = (\theta_1, \ldots, \theta_k)$,
  - covariance matrix $\Lambda$ (assumed known, but can be relaxed).

- **GOAL:** find $x^* = \text{argmax}_x \{\theta_x\}$.

- We believe that the means of the alternatives are **correlated**, that is, similar alternatives often have similar performance.
Our Problem Settings

- We consider DOvS over a finite set of alternatives $1, 2, \ldots, k$.

- Our simulator allows correlated output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector $\theta = (\theta_1, \ldots, \theta_k)$,
  - covariance matrix $\Lambda$ (assumed known, but can be relaxed).

- GOAL: find $x^* = \arg\max_x \{\theta_x\}$.

- We believe that the means of the alternatives are correlated, that is, similar alternatives often have similar performance.
  We model this belief by a multivariate normal prior on $\theta$, $\theta \sim \mathcal{N}(\mu_0, \Sigma_0)$. 
Our Problem Settings

- We consider DoVSoS over a finite set of alternatives 1, 2, ..., k.

- Our simulator allows correlated output through common random numbers (CRN). If we sample all alternatives together using CRN, we observe a normal random vector with
  - unknown mean vector $\theta = (\theta_1, \ldots, \theta_k)$,
  - covariance matrix $\Lambda$ (assumed known, but can be relaxed).

- GOAL: find $x^* = \arg\max_x \{\theta_x\}$.

- We believe that the means of the alternatives are correlated, that is, similar alternatives often have similar performance.
  We model this belief by a multivariate normal prior on $\theta$, $\theta \sim \mathcal{N}(\mu_0, \Sigma_0)$.

- We calculate VOI to decide which subset of alternatives to sample.
Related Studies

- Jones, Schonlau and Welch 1998
- Chick and Inoue 2001
- Ginsbourger, Riche and Carraro 2007
- Brochu, Cora and Freitas 2009
- Frazier, Powell and Dayanik 2009
- Negoescu, Frazier and Powell 2011
- Scott, Frazier, and Powell 2011
- Clark and Yang 1986
- Yang and Nelson 1991
- Nelson and Matejcik 1995
- Nakayama 2000
- Kim 2005
- Fu, Hu, Chen, and Xiong 2007
- Goldsman, Marshall, Kim and Nelson 2000
Related Studies

- Jones, Schonlau and Welch 1998
- Chick and Inoue 2001
- Ginsbourger, Riche and Carraro 2007
- Brochu, Cora and Freitas 2009
- Frazier, Powell and Dayanik 2009
- Negoescu, Frazier and Powell 2011
- Scott, Frazier, and Powell 2011
- Clark and Yang 1986
- Yang and Nelson 1991
- Nelson and Matejcik 1995
- Nakayama 2000
- Kim 2005
- Fu, Hu, Chen, and Xiong 2007
- Goldsman, Marshall, Kim and Nelson 2000

Novelty of Our Work:

- Correlation types: BOTH
Related Studies

- Jones, Schonlau and Welch 1998
- Chick and Inoue 2001
- Ginsbourger, Riche and Carraro 2007
- Brochu, Cora and Freitas 2009
- Frazier, Powell and Dayanik 2009
- Negoescu, Frazier and Powell 2011
- Scott, Frazier, and Powell 2011
- Clark and Yang 1986
- Yang and Nelson 1991
- Nelson and Matejcik 1995
- Nakayama 2000
- Kim 2005
- Fu, Hu, Chen, and Xiong 2007
- Goldsman, Marshall, Kim and Nelson 2000

Novelty of Our Work:

- Correlation types: BOTH
- Sampling plan: PAIRS
Related Studies

- Jones, Schonlau and Welch 1998
- Chick and Inoue 2001
- Ginsbourger, Riche and Carraro 2007
- Brochu, Cora and Freitas 2009
- Frazier, Powell and Dayanik 2009
- Negoescu, Frazier and Powell 2011
- Scott, Frazier, and Powell 2011
- Clark and Yang 1986
- Yang and Nelson 1991
- Nelson and Matejcik 1995
- Nakayama 2000
- Kim 2005
- Fu, Hu, Chen, and Xiong 2007
- Goldsman, Marshall, Kim and Nelson 2000

Novelty of Our Work:

- Correlation types: BOTH
- Sampling plan: PAIRS
- Problem Scale: LARGE
Generic Sampling Algorithm

Choose one or several alternatives to evaluate at time 1.

At each time $n = 1, 2, \ldots$ (while the stopping criterion is not met):

- Calculate the Bayesian posterior distribution on $\theta$, which is $\mathcal{N}(\mu_n, \Sigma_n)$. 

Upon stopping, report the alternative with the best estimated value as the "implementation decision".
Generic Sampling Algorithm

Choose one or several alternatives to evaluate at time 1.

At each time $n = 1, 2, \ldots$ (while the stopping criterion is not met):

- Calculate the Bayesian posterior distribution on $\theta$, which is $\mathcal{N}(\mu_n, \Sigma_n)$.
- Use this posterior to calculate the potential value of sampling certain subsets of alternatives using CRN at time $n + 1$.
- Choose a subset of alternatives $X_{n+1}$ to sample at time $n + 1$ using CRN.
Choose one or several alternatives to evaluate at time 1.

At each time $n = 1, 2, \ldots$ (while the stopping criterion is not met):

- Calculate the Bayesian posterior distribution on $\theta$, which is $\mathcal{N}(\mu_n, \Sigma_n)$.
- Use this posterior to calculate the potential value of sampling certain subsets of alternatives using CRN at time $n + 1$.
- Choose a subset of alternatives $X_{n+1}$ to sample at time $n + 1$ using CRN.

The stopping criterion is often “stop after $N$ samples”, or an adaptive rule.
Generic Sampling Algorithm

Choose one or several alternatives to evaluate at time 1.

At each time $n = 1, 2, \ldots$ (while the stopping criterion is not met):

- Calculate the Bayesian posterior distribution on $\theta$, which is $\mathcal{N}(\mu_n, \Sigma_n)$.
- Use this posterior to calculate the potential value of sampling certain subsets of alternatives using CRN at time $n + 1$.
- Choose a subset of alternatives $X_{n+1}$ to sample at time $n + 1$ using CRN.

The stopping criterion is often “stop after $N$ samples”, or an adaptive rule.

Upon stopping, report the alternative with the best estimated value as the “implementation decision”.
Value of Information

At time $n$, the VOI of evaluating subset $X$ at time $n + 1$ is

$$V_n(X) = \mathbb{E}_n \left[ \max_x \mu_{n+1,x} \mid X_{n+1} = X \right] - \max_x \mu_{n,x}.$$
Value of Information

At time $n$, the VOI of evaluating subset $X$ at time $n + 1$ is

$$V_n(X) = \mathbb{E}_n \left[ \max_x \mu_{n+1,x} \mid X_{n+1} = X \right] - \max_x \mu_{n,x}.$$  

- $\mu_{n,x}$ is the expected value of $\theta_x$ given what we know at time $n$. 

Can be computed analytically (using algorithm 1 in Frazier et al. 2009) when $X = x(\text{a singleton})$ OR $X = x(1) - x(2)$ (difference between a pair).
Value of Information

At time $n$, the VOI of evaluating subset $X$ at time $n+1$ is

$$V_n(X) = \mathbb{E}_n \left[ \max_x \mu_{n+1,x} \mid X_{n+1} = X \right] - \max_x \mu_{n,x}.$$ 

- $\mu_{n,x}$ is the expected value of $\theta_x$ given what we know at time $n$.
- $\max_x \mu_{n,x}$ is the best we can do given what we know at time $n$. 
Value of Information

At time $n$, the VOI of evaluating subset $X$ at time $n+1$ is

$$V_n(X) = \mathbb{E}_n \left[ \max_x \mu_{n+1,x} \mid X_{n+1} = X \right] - \max_x \mu_{n,x}.$$ 

- $\mu_{n,x}$ is the expected value of $\theta_x$ given what we know at time $n$.
- $\max_x \mu_{n,x}$ is the best we can do given what we know at time $n$.
- $\max_x \mu_{n+1,x}$ is the best we will be able to do given what we know at time $n$ and what we learn from the measurements at time $n+1$. 

Value of Information

At time $n$, the VOI of evaluating subset $X$ at time $n + 1$ is

$$V_n(X) = \mathbb{E}_n \left[ \max_x \mu_{n+1,x} \bigg| X_{n+1} = X \right] - \max_x \mu_{n,x}.$$ 

- $\mu_{n,x}$ is the expected value of $\theta_x$ given what we know at time $n$.
- $\max_x \mu_{n,x}$ is the best we can do given what we know at time $n$.
- $\max_x \mu_{n+1,x}$ is the best we will be able to do given what we know at time $n$ and what we learn from the measurements at time $n + 1$.

$V_n(X)$ is the expected improvement that evaluating $X$ can produce in the best estimated overall value from time $n$ to time $n + 1$. 
Value of Information

At time \( n \), the VOI of evaluating subset \( X \) at time \( n + 1 \) is

\[
V_n(X) = \mathbb{E}_n \left[ \max_x \mu_{n+1,x} \mid X_{n+1} = X \right] - \max_x \mu_{n,x}.
\]

- \( \mu_{n,x} \) is the expected value of \( \theta_x \) given what we know at time \( n \).
- \( \max_x \mu_{n,x} \) is the best we can do given what we know at time \( n \).
- \( \max_x \mu_{n+1,x} \) is the best we will be able to do given what we know at time \( n \) and what we learn from the measurements at time \( n + 1 \).

\( V_n(X) \) is the **expected improvement** that evaluating \( X \) can produce in the best estimated overall value from time \( n \) to time \( n + 1 \).

Can be computed analytically (using algorithm 1 in Frazier et al. 2009) when

\[
X = x \text{ (a singleton)} \quad \text{OR} \quad X = x^{(1)} - x^{(2)} \text{ (difference between a pair)}
\]
Knowledge Gradient Based Algorithms

At time $n$: the *Knowledge Gradient (KG) factor* of sampling $X$ next, is

$$\nu_n^{KG}(X) = \frac{V_n(X)}{c(X)},$$

where $c(X)$ is the computation cost for sampling $X$. 
Knowledge Gradient Based Algorithms

At time \( n \): the *Knowledge Gradient (KG) factor* of sampling \( X \) next, is

\[
\nu_n^{KG}(X) = \frac{V_n(X)}{c(X)},
\]

where \( c(X) \) is the computation cost for sampling \( X \).

**Algorithms** for choosing \( X_{n+1} \):

- **KG**: sample the alternative \( x \) with the largest KG factor.
  
Knowledge Gradient Based Algorithms

At time $n$: the \textit{Knowledge Gradient (KG) factor} of sampling $X$ next, is

$$\nu_{n}^{KG}(X) = \frac{V_{n}(X)}{c(X)},$$

where $c(X)$ is the computation cost for sampling $X$.

**Algorithms** for choosing $X_{n+1}$:

- **KG**: sample the alternative $x$ with the largest KG factor.

- **KG$^2$**: check the KG factors of all singletons $x$ and pairs $(x^1, x^2)$, and sample the one(s) with the largest factor.
Illustration of the KG Algorithm

\[ k = 100. \]
\[ \Lambda(i, i) = 50, \]
\[ \Lambda(i, j) = 25, \text{ for } i, j = 1, \ldots, 100. \]
\[ \mu_0 = \vec{0}, \]
\[ \Sigma_0(i, j) = 100 \exp \left[ -\frac{(i - j)^2}{50} \right]. \]
Illustration of the KG Algorithm

- $k = 100$.
- $\Lambda(i, i) = 50$, $\Lambda(i, j) = 25$, for $i, j = 1, \ldots, 100$.
- $\mu_0 = \vec{0}$,
- $\Sigma_0(i, j) = 100 \exp \left[ -\frac{(i-j)^2}{50} \right]$. 
Illustration of the KG Algorithm

\[ \log[v_{KG}^1(x)] \]

\[ n = 1 \]

- \( \theta \)
- \( \mu_1 \)
- \( \mu_1 \pm 1.96\sigma_1 \)
- \( (x_1, y_1) \)
Illustration of the KG Algorithm

\[ \theta \]

\[ \mu_1 \]

\[ \mu_1 \pm 1.96 \sigma_1 \]

\( (x_2, y_2) \)

\( (x_1, y_1) \)
Illustration of the KG Algorithm

\[\log [ v_{KG}^{n-1} (x) ]\]

- \[\theta\]
- \[\mu_{n-1}\]
- \[\mu_{n-1} \pm 1.96 \sigma_{n-1}\]
- \{(x_i, y_i)\}_{i<n}

\[\{ (x_n, y_n) \}\]
Sampling Algorithm

Illustration of the KG Algorithm

\[ n = 4 \]

\[ \log [ v_{n-1}^{KG}(x) ] \]

\[ \theta \]

\[ \mu_{n-1} \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (x_n, y_n) \]

\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG}(x) ] \]

\[ n = 5 \]

\[ \theta \]

\[ \mu_{n-1} \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (x_n, y_n) \]

\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[
\log [ v_{n-1}^{KG}(x) ]
\]

\[
\theta 
\mu_{n-1} 
\mu_{n-1} \pm 1.96 \sigma_{n-1}
(x_n, y_n)
\{(x_i, y_i)\}_{i<n}
\]

\[n = 6\]
Illustration of the KG Algorithm

\[
\log[v_{n-1}^{KG}(x)]
\]

\[
\theta
\mu_{n-1}
\mu_{n-1} \pm 1.96 \sigma_{n-1}
(x_n, y_n)
\{ (x_i, y_i) \}_{i<n}
\]

\[
\begin{align*}
&n = 7 \\
&\begin{array}{c}
\text{red} & \theta \\
\text{blue} & \mu_{n-1} \\
\text{dashed blue} & \mu_{n-1} \pm 1.96 \sigma_{n-1} \\
\text{black circle} & (x_n, y_n) \\
\text{black dots} & \{ (x_i, y_i) \}_{i<n}
\end{array}
\end{align*}
\]
Illustration of the KG Algorithm

\[ \log[v^\text{KG}_{n-1}(x)] \]

\[ n = 8 \]

\[ \theta \]

\[ \mu_{n-1} \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (x_n, y_n) \]

\[ \{(x_i, y_i)\}_{i<n} \]
Illustration of the KG Algorithm

\[ \log[v_{n-1}^{KG}(x)] \]

\[ \theta \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ \{(x_i, y_i)\}_{i<n} \]
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG}(x) ] \]

0 10 20 30 40 50 60 70 80 90 100
−60
−50
−40
−30
−20
−10

\[ \theta \]
\[ \mu_{n-1} \]
\[ \mu_{n-1} \pm 1.96 \sigma_{n-1} \]
\[ (x_n, y_n) \]
\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{\text{KG}}(x) ] \]

\[ n = 12 \]

\[ \theta \]
\[ \mu_{n-1} \]
\[ \mu_{n-1} \pm 1.96 \sigma_{n-1} \]
\[ (x_n, y_n) \]
\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[
\log [ v_{n-1}^{KG}(x) ]
\]

\[
\theta, \mu_{n-1}, \mu_{n-1} \pm 1.96\sigma_{n-1}, (x_n, y_n), \{ (x_i, y_i) \}_{i<n}
\]
Illustration of the KG Algorithm

\[ \log[v_{n-1}^{KG}(x)] \]

\[ \theta \]

\[ \mu_{n-1} \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (x_n, y_n) \]

\[ \{(x_i, y_i)\}_{i<n} \]
Illustration of the KG Algorithm

\[ \theta \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ \{(x_i, y_i)\}_{i<n} \]
Illustration of the KG Algorithm

\[ \log[v_{n-1}^{KG}(x)] \]

\[ n = 20 \]

\[ \theta \]

\[ \mu_{n-1} \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (x_n, y_n) \]

\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[ n = 22 \]

\[ \log [ v_{n-1}^{KG}(x) ] \]
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG}(x) ] \]

\[ \theta \]

\[ \mu_{n-1} \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (x_n, y_n) \]

\[ \{(x_i, y_i)\}_{i<n} \]
Illustration of the KG Algorithm

\[ \log \left[ v_{n-1}^{KG} (x) \right] \]

- Red: \( \theta \)
- Blue: \( \mu_{n-1} \)
- Dashed blue: \( \mu_{n-1} \pm 1.96\sigma_{n-1} \)
- Black circle: \( (x_n, y_n) \)
- Gray dots: \( \{ (x_i, y_i) \}_{i<n} \)
Illustration of the KG Algorithm

\begin{align*}
\log [ v_{n-1}^{\text{KG}}(x) ]
\end{align*}
Illustration of the KG Algorithm

\[
\log [ v_{n-1}^{KG}(x) ]
\]

\[n = 40\]

- \(\theta\)
- \(\mu_{n-1}\)
- \(\mu_{n-1} \pm 1.96\sigma_{n-1}\)
- \((x_n, y_n)\)
- \(\{(x_i, y_i)\}_{i<n}\)
Illustration of the KG Algorithm

\[ n = 45 \]

\[ \log [ v_{n-1}^{KG} (x) ] \]

\[ \theta \]
\[ \mu_{n-1} \]
\[ \mu_{n-1} \pm 1.96 \sigma_{n-1} \]
\[ (x_n, y_n) \]
\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG}(x) ] \]

\[ n = 50 \]

\[ \theta \]
\[ \mu_{n-1} \]
\[ \pm 1.96 \sigma_{n-1} \]

\( (x_n, y_n) \)
\( \{ (x_i, y_i) \}_{i<n} \)
Illustration of the KG Algorithm

\[ \log[v_{n-1}^{KG}(x)] \]

\[ n = 60 \]

\[ \theta \]

\[ \mu_{n-1} \]

\[ \mu_{n-1} \pm 1.96 \sigma_{n-1} \]

\[ (x_n, y_n) \]

\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG}(x) ] \]

\( n = 70 \)

\[ \theta \]
\[ \mu_{n-1} \]
\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]
\[ (x_n, y_n) \]
\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG}(x) ] \]

\[ n = 80 \]

\[ \theta, \mu_{n-1}, \mu_{n-1} \pm 1.96 \sigma_{n-1}, (x_n, y_n), \{(x_i, y_i)\}_{i<n} \]
Illustration of the KG Algorithm

\[ \log[v_{n-1}^{KG}(x)] \]

\[ n = 90 \]

\[ \theta \]
\[ \mu_{n-1} \]
\[ \mu_{n-1} \pm 1.96 \sigma_{n-1} \]
\[ (x_n, y_n) \]
\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[ n = 100 \]

\[
\log [ v^{KG}_{n-1}(x) ]
\]

\[ \theta \quad \mu_{n-1} \quad \mu_{n-1} \pm 1.96\sigma_{n-1} \quad (x_n, y_n) \quad \{(x_i, y_i)\}_{i<n} \]
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG}(x) ] \]

\[ n = 120 \]
Illustration of the KG Algorithm

\( n = 140 \)

\[ \log [ v_{n-1}^{KG}(x) ] \]

- \( \theta \)
- \( \mu_{n-1} \)
- \( \mu_{n-1} \pm 1.96 \sigma_{n-1} \)
- \( (x_n, y_n) \)
- \( \{ (x_i, y_i) \}_{i<n} \)
Illustration of the KG Algorithm

\[ \log[v_{n-1}(x)] \]

\( n = 160 \)

\[ \theta \]

\[ \mu_{n-1} \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (x_n, y_n) \]

\[ \{ (x_i, y_i) \}_{i<n} \]
Illustration of the KG Algorithm

\[ \log[v_{n-1}^{KG}(x)] \]

\[ n = 180 \]

\[ \theta \]
\[ \mu_{n-1} \]
\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]
\[ (x_n, y_n) \]
\[ \{(x_i, y_i)\}_{i<n} \]
Illustration of the KG Algorithm

$n = 200$

$\theta$

$\mu_{n-1}$

$\mu_{n-1} \pm 1.96 \sigma_{n-1}$

$(x_n, y_n)$

$\{ (x_i, y_i) \}_{i<n}$

$\log [ v_{n-1}^{KG}(x) ]$
Illustration of the KG Algorithm

\begin{align*}
n = 220
\end{align*}

\begin{align*}
\log \left[ v_{n-1}^{KG}(x) \right]
\end{align*}
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG} (x) ] \]

- \( n = 240 \)
- \( \theta, \mu_{n-1}, \mu_{n-1} \pm 1.96 \sigma_{n-1} \)
- \( \{ (x_i, y_i) \}_{i<n} \)
Illustration of the KG Algorithm

\[ \log[v_{n-1}^{KG}(x)] \]

\( n = 260 \)

\[ \theta \]
\[ \mu_{n-1} \]
\[ \mu_{n-1} \pm 1.96 \sigma_{n-1} \]

\( (x_n, y_n) \)
\( \{(x_i, y_i)\}_{i<n} \)
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG}(x) ] \]
Illustration of the KG Algorithm

\[ \text{log} \left[ v^{KG}_{n-1}(x) \right] \]
Illustration of the KG Algorithm

\[ \log \left[ v_{n-1}^{KG} (x) \right] \]
Illustration of the KG Algorithm
Illustration of the KG Algorithm

\[ \log [ v_{n-1}^{KG} (x) ] \]

\[ \theta, \mu_{n-1}, \mu_{n-1} \pm 1.96 \sigma_{n-1} \]

\[ (x_n, y_n), \{(x_i, y_i)\}_{i<n} \]
Illustration of the KG Algorithm

\begin{align*}
n = 380 \\
\log[v_{n-1}^{\text{KG}}(x)] \\
\theta = \mu_{n-1} \pm 1.96 \sigma_{n-1}
\end{align*}
Illustration of the KG$^2$ Algorithm
Illustration of the $\text{KG}^2$ Algorithm

\[
\log [ v_{n-1}^{\text{KG}} ( x^{(1)}, x^{(2)} ) ]
\]

\[
\theta \quad \mu_{n-1} \quad \mu_{n-1} \pm 1.96 \sigma_{n-1} (x_n, y_n)
\]

{(X_i, Y_i)}_{i<n}
Illustration of the $KG^2$ Algorithm

\[
\log [v_{n-1}(x^{(1)}, x^{(2)}))] = 3
\]

$\theta, \mu_{n-1}, \mu_{n-1} \pm 1.96\sigma_{n-1}, (X_n, Y_n), \{ (X_i, Y_i) \}_{i<n}$
Illustration of the KG$^2$ Algorithm

\[ \log [ v_{n-1}^{KG}(x^{(1)}, x^{(2)}) ] \]

\[ \theta \pm 1.96 \sigma_{n-1}(x_n, y_n) \]

\[ \{(X_i, Y_i)\}_{i<n} \]
Illustration of the $KG^2$ Algorithm
Illustration of the $KG^2$ Algorithm

$$\log[v_{n-1}(x^{(1)}, x^{(2)})]$$
Illustration of the $KG^2$ Algorithm

\[ \log[v_{n-1}(x^{(1)}, x^{(2)})] \]

\[ \theta, \mu_{n-1}, \mu_{n-1} \pm 1.96\sigma_{n-1}, (x_n, y_n), \{(X_i, Y_i)\}_{i<n} \]
Illustration of the \( KG^2 \) Algorithm

\[
\log [ v_{KG}^{n-1}(x^{(1)}, x^{(2)}) ]
\]

\[
\theta
\mu_{n-1}
\mu_{n-1} \pm 1.96\sigma_{n-1}
(x_n, y_n)
\{(X_i, Y_i)\}_{i<n}
\]
Illustration of the $KG^2$ Algorithm

$n = 9$

$\theta$
$\mu_{n-1}$
$\mu_{n-1} \pm 1.96\sigma_{n-1}$
$(x_n, y_n)$
$(X_i, Y_i)_{i<n}$

$\log[v_{n-1}(x^{(1)}, x^{(2)})]$
Illustration of the KG\(^2\) Algorithm

\[
\log [ v_{n-1}^{KG} (x^{(1)}, x^{(2)}) ]
\]

\[
\theta \quad \mu_{n-1}^t \quad \mu_{n-1}^t \pm 1.96 \sigma_{n-1}^t
\]

\[
(X_n, Y_n) \quad \{ (X_i, Y_i) \}_{i<n}
\]
Illustration of the KG² Algorithm

\[
\log \left[ v_{n-1}^{KG} \left( x^{(1)}, x^{(2)} \right) \right]
\]

\[
n = 11
\]

\[
\theta \quad \mu_{n-1} \quad \mu_{n-1} \pm 1.96\sigma_{n-1} \quad (X_n, Y_n) \quad \{(X_i, Y_i)\}_{i<n}
\]
Illustration of the $KG^2$ Algorithm
Illustration of the $KG^2$ Algorithm

\[
\log [v_{n-1}^{KG} (x^{(1)}, x^{(2)}) ]
\]
Illustration of the $KG^2$ Algorithm

$n = 14$

$\theta$

$\mu_{n-1}$

$\mu_{n-1} \pm 1.96\sigma_{n-1}$

$(X_n, Y_n)$

$\{(X_i, Y_i)\}_{i<n}$

$\log [v_{n-1}^{KG}(x^{(1)}, x^{(2)})]$
Illustration of the $KG^2$ Algorithm

\[ \log [v_{n-1}(x^{(1)}, x^{(2)})] \]

\[ n = 15 \]

\[ \theta \]

\[ \mu_{n-1} \pm 1.96 \sigma_{n-1} \]

\[ (X_n, Y_n) \]

\[ \{(X_i, Y_i)\}_{i<n} \]
Illustration of the $KG^2$ Algorithm
Illustration of the KG\(^2\) Algorithm

\[ \log [ v_{n-1}^{KG} (X_n, Y_n) ] \]

\[ \theta \]

\[ \mu_{n-1} \pm 1.96 \sigma_{n-1} \]

\[ (X_n, Y_n) \]

\[ \{ (X_i, Y_i) \}_{i<n} \]
Illustration of the $K G^2$ Algorithm

\[ \log[v_{n-1}(x^{(1)}, x^{(2)})] \]

\[ n = 18 \]
Illustration of the $KG^2$ Algorithm

\[ \log [v_{n-1}^{KG}(x^{(1)}, x^{(2)})] \]

$n = 19$

\[ \theta_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (X_n, Y_n) \]

\[ \{(X_i, Y_i)\}_{i<n} \]
Illustration of the $KG^2$ Algorithm

$$\log[v_{n-1}^{KG}(x^{(1)}, x^{(2)})]$$

$n = 20$

$\theta$
$\mu_{n-1}$
$\mu_{n-1} \pm 1.96\sigma_{n-1}$
$(X_n, Y_n)$
${(X_i, Y_i)}_{i<n}$
Illustration of the $KG^2$ Algorithm
Illustration of the $KG^2$ Algorithm

$n = 24$

$log [ v_{n-1}^{KG}(x^{(1)}, x^{(2)}) ]$

\[ n = 24 \]

\[ \log [ v_{n-1}^{KG}(x^{(1)}, x^{(2)}) ] \]
Illustration of the KG² Algorithm

\[
\log [ v_{n-1}^{KG}(x^{(1)}, x^{(2)}) ]
\]
Illustration of the $KG^2$ Algorithm

\[
\log [ v_{n-1}^{KG} (x^{(1)}, x^{(2)}) ]
\]

\[
\theta
\mu_{n-1}
\mu_{n-1} \pm 1.96 \sigma_{n-1}
(x_n, y_n)
(X_i, Y_i)_{i<n}
\]
Illustration of the $KG^2$ Algorithm
Illustration of the $KG^2$ Algorithm
Illustration of the $KG^2$ Algorithm
Illustration of the $\text{KG}^2$ Algorithm
Illustration of the $KG^2$ Algorithm

\[ \log[v_{n-1}(x^{(1)}, x^{(2)})] \]

\[ n = 38 \]

\[ \theta, \mu_{n-1}, \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (x_n, y_n) \]

\[ \{(X_i, Y_i)\}_{i<n} \]
Illustration of the $KG^2$ Algorithm
Illustration of the $KG^2$ Algorithm

\[ \log[v_{n-1}(x^{(1)}, x^{(2)})] = \theta + \mu_{n-1} \pm 1.96 \sigma_{n-1} \]

\( n = 45 \)

\[ (X_i, Y_i)_{i<n} \]
Illustration of the KG² Algorithm
Illustration of the KG^2 Algorithm

\[ \log [ v_{n-1}^{KG}(x^{(1)}, x^{(2)}) ] \]

\[ n = 55 \]

\[ \theta \]

\[ \mu_{n-1} \]

\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (X_n, Y_n) \]

\[ \{(X_i, Y_i)\}_{i<n} \]
Illustration of the $KG^2$ Algorithm

![Illustration of the $KG^2$ Algorithm](image)

\[ \log [ v_{n-1}^{KG}(x^{(1)}, x^{(2)}) ] \]

- $n = 60$
- $x^{(1)}$
- $x^{(2)}$
- $\log [ v_{n-1}^{KG}(x^{(1)}, x^{(2)}) ]$

\[ \theta \]
\[ \mu_{n-1} \]
\[ \mu_{n-1} \pm 1.96\sigma_{n-1} \]
\[ (X_n, Y_n) \]
\[ \{(X_i, Y_i)\}_{i<n} \]
Illustration of the $KG^2$ Algorithm

\[ \log [ v_{n-1}^{KG}(x^{(1)}, x^{(2)}) ] \]

\[ n = 65 \]

\[ \theta \]

\[ \mu_{n-1} \pm 1.96 \sigma_{n-1} \]

\[ (X_n, Y_n) \]

\[ \{(X_i, Y_i)\}_{i<n} \]
Illustration of the $KG^2$ Algorithm

\[ \log[v_{n-1}(x^{(1)}, x^{(2)})] \]

\[ n = 70 \]

\[ \theta \quad \mu_{n-1} \quad \mu_{n-1} \pm 1.96\sigma_{n-1} \]

\[ (X_n, Y_n) \quad \{(X_i, Y_i)\}_{i<n} \]
Illustration of the $KG^2$ Algorithm

$n = 75$

$log [ v_{KG}^{n-1}(x^{(1)}, x^{(2)}) ]$

$\theta$
$\mu_{n-1}$
$\mu_{n-1} \pm 1.96\sigma_{n-1}$
$(X_n, Y_n)$
$\{(X_i, Y_i)\}_{i<n}$

$\theta$
$\mu_{n-1}$
$\mu_{n-1} \pm 1.96\sigma_{n-1}$
$(X_n, Y_n)$
$\{(X_i, Y_i)\}_{i<n}$
Illustration of the $KG^2$ Algorithm
Illustration of the KG^2 Algorithm

\[
\log[v_{n-1}(x^{(1)}, x^{(2)})]
\]

\[
\theta_{n}, \mu_{n-1} \pm 1.96\sigma_{n-1}(X_n, Y_n)
\]

\[
\{ (X_i, Y_i) \}_{i<n}
\]
Illustration of the $\text{KG}^2$ Algorithm

\[ \log [v_{n-1}(x^{(1)}, x^{(2)})] \]
Illustration of the $KG^2$ Algorithm
Illustration of the KG² Algorithm

\[ \log [ v_{n-1}^{KG} (x^{(1)}, x^{(2)}) ] \]
Exploitation vs. Exploration

- KG factor is \textit{bigger} when the \textit{posterior mean} is \textit{bigger}.

- KG factor is \textit{bigger} when the \textit{posterior variance} is \textit{bigger}.
Exploitation vs. Exploration

- KG factor is \textit{bigger} when the \textit{posterior mean} is \textit{bigger}.

- KG factor is \textit{bigger} when the \textit{posterior variance} is \textit{bigger}.

- These two tendencies often push against each other, and our sampling algorithms manage to \text{\textsc{balance}} them.
What if the Solution Space is BIG?

\[ \text{VOI?} \]

\[ V_n(X) = \mathbb{E}_n[\max_{x} \mu_{n+1}, x | x_{n+1} = X - \max_{x} \mu_{n}, x] \approx \mathbb{E}_n[\max_{x \in s_{ss}} \mu_{n+1}, x | x_{n+1} = X - \max_{x \in s_{ss}} \mu_{n}, x] \]

where \( s_{ss} = \) some small subset.
What if the Solution Space is BIG?

- VOI?

\[
V_n(X) = \mathbb{E}_n \left[ \max_{x} \mu_{n+1,x} \mid X_{n+1} = X \right] - \max_{x} \mu_{n,x}
\]

\[
\approx \mathbb{E}_n \left[ \max_{x \in sss} \mu_{n+1,x} \mid X_{n+1} = X \right] - \max_{x \in sss} \mu_{n,x},
\]

where \( sss \) = some small subset.
What if the Solution Space is BIG?

- VOI?

\[ V_n(X) = \mathbb{E}_n \left[ \max_x \mu_{n+1,x} \mid X_{n+1} = X \right] - \max_x \mu_{n,x} \]

\[ \approx \mathbb{E}_n \left[ \max_{x \in sss} \mu_{n+1,x} \mid X_{n+1} = X \right] - \max_{x \in sss} \mu_{n,x}, \]

where \( sss = \) some small subset.

- Sampling Decision?

\[ \arg\max_{X \in \ldots} \nu^KG_n(X) \leftrightarrow \text{multi-start gradient decent} \]
What if the Solution Space is BIG?

• VOI?

\[ V_n(X) = \mathbb{E}_n \left[ \max_x \mu_{n+1,x} \left| X_{n+1} = X \right. \right] - \max_x \mu_{n,x} \]

\[ \approx \mathbb{E}_n \left[ \max_{x \in \text{sss}} \mu_{n+1,x} \left| X_{n+1} = X \right. \right] - \max_{x \in \text{sss}} \mu_{n,x}, \]

where \( \text{sss} \) = some small subset.

• Sampling Decision?

\[ \arg\max_{X \in \ldots} \nu^{KG}_n(X) \Leftarrow \text{multi-start gradient decent} \]

• Implementation Decision?

\[ \arg\max_{X \in \text{SAMPLED}} \mu_{n,x} \approx \arg\max_{x \in \text{SAMPLED}} \mu_{n,x} \]
Why Knowledge Gradient?

We have replaced one optimization problem: \( \max_x \theta_x \) with many optimization problems: \( \max_X \nu_n^{KG}(X) \), for \( n = 1, 2, \ldots \).

WHY is this a good thing?
Why Knowledge Gradient?

We have replaced one optimization problem: \( \max_x \theta_x \) with many optimization problems: \( \max_X \nu_n^{KG}(X) \), for \( n = 1, 2, \ldots \)

WHY is this a good thing?

- Evaluating \( \theta_x \) is expensive (minutes, hours, days), and derivative information is unavailable.
Why Knowledge Gradient?

We have replaced one optimization problem: \( \max_x \theta_x \) with many optimization problems: \( \max_X \nu^{KG}_{n}(X) \), for \( n = 1, 2, \ldots \).

WHY is this a good thing?

- Evaluating \( \theta_x \) is expensive (minutes, hours, days), and derivative information is unavailable.
- Evaluating \( \nu^{KG}_{n}(X) \) is quick (microseconds), and derivative information is available.
Why Knowledge Gradient?

We have replaced one optimization problem: $\max_x \theta_x$ with many optimization problems: $\max_X \nu^KG_n(X)$, for $n = 1, 2, \ldots$

WHY is this a good thing?

- Evaluating $\theta_x$ is expensive (minutes, hours, days), and derivative information is unavailable.
- Evaluating $\nu^KG_n(X)$ is quick (microseconds), and derivative information is available.
- We spend longer to decide where to take each sample, but require much fewer samples to find a good solution!
Rosenbrock with $10^6$ alternatives

- **RSGP**: random search with a correlated (Gaussian Process) prior.
- **Opportunity Cost** $= \max_x \theta_x - \theta_{x_n^*}$, where $x_n^* = \arg\max_x \mu_{n,x}$. 

The GP prior & the sampling covariances (assumed unknown) are estimated by Maximum Likelihood.
Rosenbrock with $10^6$ alternatives

- **RSGP**: random search with a correlated (Gaussian Process) prior.
- **Opportunity Cost** = \( \max_x \theta_x - \theta_{x^*_n} \), where \( x^*_n = \arg\max_{x \in \text{sampled}} \mu_{n,x} \).
- The GP prior & the sampling covariances (assumed unknown) are estimated by *Maximum Likelihood*. 
Rosenbrock with $10^6$ alternatives

- **RSGP**: random search with a correlated (Gaussian Process) prior.
- **Opportunity Cost** = $\max_x \theta_x - \theta_{x^*_n}$, where $x^*_n = \arg\max_{x \in \text{sampled}} \mu_{n,x}$.
- The GP prior & the sampling covariances (assumed unknown) are estimated by Maximum Likelihood.
- $\text{KG}^2 \succ \text{KG} \succ \text{RS}$
Rosenbrock with $10^6$ alternatives

- **RSGP**: random search with a correlated (Gaussian Process) prior.
- **Opportunity Cost** $= \max_x \theta_x - \theta_{x^*_n}$, where $x^*_n = \arg\max_{x \in \text{sampled}} \mu_{n,x}$.
- The GP prior & the sampling covariances (assumed unknown) are estimated by *Maximum Likelihood*.

**Results**:
- $\text{KG}^2 \succ \text{KG} \succ \text{RS}$
- Correlated prior $\succ$ independent prior
Assemble to Order problem with $21^8$ alternatives

- **Industrial Strength COMPASS (ISC):** Xu et al. 2010

![Graph showing Estimated Profit vs Sample Size](image)
Assemble to Order problem with $21^8$ alternatives

- Industrial Strength COMPASS (ISC): Xu et al. 2010

KG/$KG^2$ spends 10 times longer than ISC to compute the 1000 sampling decisions.
Assemble to Order problem with $21^8$ alternatives

- Industrial Strength COMPASS (ISC): Xu et al. 2010

KG/ KG$^2$ spends 10 times longer than ISC to compute the 1000 sampling decisions.

**BUT**

- ISC takes $1000+$ samples / $20+$ minutes on average to reach an average profit of 115.
Assemble to Order problem with $21^8$ alternatives

- Industrial Strength COMPASS (ISC): Xu et al. 2010

KG/KG$^2$ spends 10 times longer than ISC to compute the 1000 sampling decisions.

BUT

- ISC takes $1000+\text{ samples} / 20+\text{ minutes}$ on average to reach an average profit of 115.
- KG requires $300-\text{ samples} / 10-\text{ minutes}$ on average to reach 115.
- KG$^2$ requires $220-\text{ samples} / 6-\text{ minutes}$ on average to reach 115.
Numerical Results

Ideal Scope: Expensive Function Evaluation

Figure: CPU time spent in a sample path of $KG^2$, on the ATO problem.

$KG/KG^2$ are less suitable for problems in which simulation can be performed very quickly.
Ideal Scope: Expensive Function Evaluation

$KG/KG^2$ are less suitable for problems in which simulation can be performed very quickly.

When samples come from a complex, long-running simulator, however,

- the substantial computational consumption in deciding where to sample is relatively unimportant;
- algorithms like $KG^2$ that find good solutions in fewer samples also work well in terms of overall computation time.

Figure: CPU time spent in a sample path of $KG^2$, on the ATO problem.
Conclusions

- We take advantage of both
  - correlated prior beliefs & correlated sampling distributions
  - in a *Bayesian value of information* framework, which brings a distinct benefit for optimization via simulation.
Conclusions

- We take advantage of both
  - correlated prior beliefs & correlated sampling distributions
  in a *Bayesian value of information* framework, which brings a distinct benefit for optimization via simulation.

- We give easy-to-verify conditions under which almost sure convergence to the optimal solution can be guaranteed.
Conclusions

- We take advantage of both
  - correlated prior beliefs & correlated sampling distributions
  in a *Bayesian value of information* framework, which brings a distinct benefit for optimization via simulation.

- We give easy-to-verify conditions under which almost sure convergence to the optimal solution can be guaranteed.

- Our algorithms demonstrate superior efficiency compared to others
  - in problems with combinatorially large solution spaces, and
  - when samples are moderately to very computationally expensive.

*Run times are a low order polynomial in the number of samples observed, rather than a low order polynomial in the size of the solution space!*
THANK YOU!


References


