Stochastic global optimization practice in nuclear criticality safety assessment

Dr. Y. Richet

Criticality Assessment Study and Research Department

17 / 05 / 2011
IHP - Paris
Overview

Nuclear criticality-safety assessment and related issues
- Classical approach and basic tools
- Rethinking through « Computer Experiments » framework:
  « Expert driven » supplemented by « Algorithm assisted» policy

Components to build an operational workbench
- Front-end GUI
- Grid computing engine & algorithm back-end

Feedback on two years of daily use
- Adhesion vs. resiliency
- Enhancing robustness with EGO/kriging improvements

Focus on stochastic optimization
- Related [R] packages
- Integration of stochastic simulator
Nuclear criticality-safety assessment and related issues
Nuclear criticality-safety assessment and related issues

Classical approach and basic tools

Nuclear criticality-safety

- Physics = neutron transport, chemistry, materials science
- System = industrial storage, transport cask, ...
- Hypothesis = environment conditions: flooding, earthquake, overloading
- Output of interest = neutrons multiplication factor (aka k-effective), criticality means k-effective > 1.0
- Safety demonstration = k-effective not to exceed 0.95, in the hypothesis range

Assessment: modeling criticality accident risk

Expert knowledge assessment ... supplemented by numerical checking:

- Physics <-> criticality simulation code
- System <-> input dataset
- Hypothesis <-> input dataset variables (dim<10)
- Output of interest <-> (scalar) output of the code
Nuclear criticality-safety assessment and related issues

Classical approach and basic tools

Parametric study for maximization of k-effective
- < 10 scalar, independent & compact parameters
- Non linear & cross effects
- (often) Several local max

(Mc) Monte Carlo simulation of k-effective
- One calculation point costs 10 s. to 1 h.
- Endpoint simulation maybe:
  - $sd(k\text{-effective}) < \text{input setting (say 0.00100)}$
  - $\text{quantile}(k\text{-effective},0.999) < \text{input setting (say 0.95000 or current max.)}$

Remote/grid calculation issue at IRSN
- ~ 80 heterogeneous CPU available
- Not (yet) parallelized code (Markov chain)
Nuclear criticality-safety assessment and related issues

Classical approach and basic tools

- Case study = interim storage of dry PuO2 powder, variables being:
  - Powder loading as powder density in [0.5, 4.0]
  - Storage flooding as water density between cans in [0.0, 1.0]
Nuclear criticality-safety assessment and related issues

Rethinking through « Computer Experiments » framework

Expert-driven approach

- OaT maximization
  - 2 or 3 parameters
  - (supposed) penalizing value for other parameters
  - 1 or 2 cycles by hand ...
- Factorial DoE
  - 5 to 10 points / dimension
  - 2 or 3 parameters
  - automated task with dedicated software (including PROMETHEE)

Case study results
Nuclear criticality-safety assessment and related issues

Rethinking through « Computer Experiments » framework

Expert-driven approach - case study results

- OaT maximization

![Diagram showing neutron multiplication factor vs. powder density](image1)

-15 pts

![Diagram showing neutron multiplication factor vs. density of water between storage tubes](image2)

-15 pts
Nuclear criticality-safety assessment and related issues

- Rethinking through « Computer Experiments » framework

Expert-driven approach - case study results

- Factorial DoE
Nuclear criticality-safety assessment and related issues

Rethinking through « Computer Experiments » framework

EGO-assisted approach

- Initial DoE
  - LHS
  - (cross) Bounds of parameters
- Kriging surrogate of noisy (as MC) experiments
- Maximization of Expected Improvement criterion
- Loop on Kriging/maxEI ...

Case study results
Nuclear criticality-safety assessment and related issues

- Rethinking through « Computer Experiments » framework

- EGO-assisted approach - case study results
Overview

Nuclear criticality-safety assessment and related issues
  ▪ Classical approach and basic tools
  ▪ Rethinking through « Computer Experiments » framework:
    « Expert driven » supplemented by « Algorithm assisted » policy

Components to build an operational workbench
  ▪ Front-end GUI
  ▪ Grid computing engine & algorithm back-end

Feedback on two years of daily use
  ▪ Adhesion vs. resiliency
  ▪ Enhancing robustness with EGO/kriging improvements

Focus on stochastic optimization
  ▪ Related [R] packages
  ▪ Integration of stochastic simulator
Components to build an operational workbench
Components to build an operational workbench

Front-end GUI

End user input
- Based on ASCII input dataset of your code (application field agnostic)
- User free to insert « $parameter » or formulas anywhere in these file
- Select any DoE policy available for this input parameters / code output
  - Factorial DoE, Efficient Global Optimization
  - ... (SA, determ. optimization, inversion, ...)

Computing
- (blindly) Launch the grid computing workflow
- (online) Follow intermediate results (if available)

Analysis
- Get results
- ... and a true applicable conclusion (or a new question :)
Components to build an operational workbench
Components to build an operational workbench

Stochastic global optimization practice in nuclear criticality safety assessment
Components to build an operational workbench
Components to build an operational workbench
Components to build an operational workbench

Stochastic global optimization practice in nuclear criticality safety assessment
Components to build an operational workbench
Components to build an operational workbench

Stochastic global optimization practice in nuclear criticality safety assessment
Components to build an operational workbench
Components to build an operational workbench

**Data sets**

**Calculations**

- Data set: Pu239, Pu239
- State: 26.21
- Activity: 0.0

**Optimum**

- Size = 22
- Maximum value is 0.87757 (sd=0.7E-4)
- d.PuO2 = 3.5
- d.broul.scale = 0

Next expected maximum value may be 0.40375585564175535 (sd=0.9512931183748922)

Improvement sequence is 0.03634326563116535 0.015487485658755436

**Results**

**Stochastic global optimization practice in nuclear criticality safety assessment**
Components to build an operational workbench
Components to build an operational workbench

Stochastic global optimization practice in nuclear criticality safety assessment
Components to build an operational workbench

**Configuration**

- Data sets
  - Data set | State | Pts | Variables
- Calculations
  - Data set | State | Activity
- Results
  - Data set | Size | Pts | State

**Post processing**

- Optimum
  - Size = 58
  - Maximum value is 0.00964 (std=9.6E-4)
  - d.PuO2 = 1.01104293735437
  - d.broul.scale = 0.093274047176015

Next expected maximum value may be 0.3413372439640373 (std=0.00697566035673094)

Improvement sequence is: 0.0263532656811525, 0.0154874856873943, 0.040748447893978, 2.763658399516977, 0.0158927121256177

**Stochastic global optimization practice in nuclear criticality safety assessment**
Components to build an operational workbench
Components to build an operational workbench

Grid computing engine & algorithm back-end

1. Distributed computing engine
   - Asynchronous, remote & parallel distribution
   - Back-end daemon compatible with larger set of CPU boxes: server, workstation, grid, cluster, ... and even Windows office desktop
   - Dynamic merge of heterogeneous computing power
   - Failover

2. Algorithm back-end
   - [R] (remote or local)
   - Light wrapping script
Components to build an operational workbench

Grid computing engine & algorithm back-end

1. Input parameters
   \[ X = \text{double}[] \]

2. Supervisor

3. Simulator

4. Algorithm

5. [R] Server

6. Output values
   \[ y = \text{double}[] \]

7. Running info

8. R::analyseDesign(X=..., y=....)

9. X <- R::nextDesign(....)
Components to build an operational workbench

Grid computing engine & algorithm back-end

```r
# constructor and initializer of R session
init <- function() {
  library(lhs)
  ...
}

# first design building. All variables are set in [0,1]. d is the dimension, or number of variables
# @param d number of variables
# @return next design of experiments
buildInitialDesign <- function(d) {
  set.seed(1)
  lhs <- maximinLHS(n=initBatchSize,k=d)
  ...
}

# iterated design building.
# @param X data frame of current doe variables (in [0,1])
# @param Y data frame of current results
# @return next doe step
prepareNextDesign <- function(X,Y) {
  ...
  return(as.matrix(Xnext))
}

# final analysis. All variables are set in [0,1]. Return HTML string
# @param X data frame of doe variables (in [0,1])
# @param Y data frame of results
# @return HTML string of analysis
analyseDesign <- function(X,Y) {
  ...
  html=paste(sep="<br/>",paste("<HTML>... ",m),...,"</HTML>")
  ...
  return(paste(html,plot))
}
Components to build an operational workbench

Grid computing engine & algorithm back-end

```r
buildInitialDesign <- function(d) {...}

#' iterated design building.
#' @param X data frame of current doe variables (in [0,1])
#' @param Y data frame of current results
#' @return next DoE step
prepareNextDesign <- function(X,Y) {
  if (iEGO > iterations) return();
  iEGO <<- iEGO + 1
  ...
  noise.var <<- as.array(Y[,2])
  if (search_min) {y=Y[,1]} else {y=-Y[,1]}
  ...
  kmi <- km(design=X, response=y, trend, optim.method='gen',...)
  EGOi <- max_qEI.CL(model=kmi, npoints=batchSize, L=liar(as.array(Y[,1])), ...)
  return(as.matrix(EGOi$par))
}

#' final analysis. All variables are set in [0,1]. Return HTML string
#' @param X data frame of doe variables (in [0,1])
#' @param Y data frame of results
#' @return HTML string of analysis
analyseDesign <- function(X,Y) {
  analyse.files <<- paste("sectionview_",iEGO-.1,".png",sep="")
  png(file=analyse.files,bg="transparent",height=resolution,width = resolution)
  ...
  html <- paste(sep="<br/>",paste("<HTML>minimum is ",m),...
```
Overview

Nuclear criticality-safety assessment and related issues
- Classical approach and basic tools
- Rethinking through « Computer Experiments » framework:
  « Expert driven » supplemented by « Algorithm assisted» policy

Components to build an operational workbench
- Front-end GUI
- Grid computing engine & algorithm back-end

Feedback on two years of daily use
- Adhesion vs. resiliency
- Enhancing robustness with EGO/kriging improvements

Focus on stochastic optimization
- Related [R] packages
- Integration of stochastic simulator
Feedback on two years of daily use
Feedback on two years of daily use

üp Adhesion vs. Resiliency

Resiliency of « old school » practices

- Coverage: Only 50-80% of assessment practice is covered by the « computer experiments » framework...
  => Let users do whatever they want (including previous old practice)
- Quality Insurance: workflow is already mapped on existing tools...
  => The workbench have to be flexible enough to permit same level of QI
- Users: to master new concepts, the learning curve is sometimes too steep
  => Take time to explain...

Adhesion to this “new” assessment framework

- <at first> focus on early adopters (easier 20% of target)
- <then> capitalize on « real world » feedback, to adapt the solution
- <try to> convince wider and wider… adapt again and again…
- <finally> involve people in R&D policy

- Efficiency measure based on « real world » test cases
- Smooth (software) transition between old and new practices
Feedback on two years of daily use

Enhancing robustness with EGO/kriging improvements

- Noisy kriging
  - Noise to model random Gaussian output of MC code
  - Heteroscedasticity to support arbitrary « fidelity » of experiments

- Automatic input « scaling »
  - Support for input parameters transformations
    log/exp/logistic emulated as local 2nd degree splines
  - To be published in DiceKriging 1.2 (soon)

- Parallel EGO
  - Constant Liar heuristic
  - Tunnable deepening of EGO
Overview

1. Nuclear criticality-safety assessment and related issues
   - Classical approach and basic tools
   - Rethinking through « Computer Experiments » framework:
     « Expert driven » supplemented by « Algorithm assisted» policy

2. Components to build an operational workbench
   - Front-end GUI
   - Grid computing engine & algorithm back-end

3. Feedback on two years of daily use
   - Adhesion vs. resiliency
   - Enhancing robustness with EGO/kriging improvements

4. Focus on stochastic optimization
   - Related [R] packages
   - Integration of stochastic simulator
Focus on stochastic optimization
Focus on stochastic optimization

**Related [R] packages**

DiceKriging
- Ordinary / Universal kriging
- Nugget / Noise

- Kernel:
  - Exponential
  - Gauss
  - Power-exponential
  - Matern 3/2
  - Matern 5/2
Focus on stochastic optimization

Related [R] packages

**DiceKriging**
- Ordinary / Universal kriging
- Nugget / Noise
- Kernel: \( C(x, y) = s^2 k(x, y) \)
  - Exponential
    \[ k|h| = |x - y| = e^{-\frac{h}{\theta}} \]
  - Gauss
    \[ k|h| = |x - y| = e^{-\frac{(h)^2}{\theta}} \]
  - Power-exponential
    \[ k|h| = |x - y| = e^{-\frac{(h)^p}{\theta}} \]
  - Matern 3/2
    \[ k|h| = |x - y| = \left(1 + \sqrt{3} \frac{h}{\theta}\right) e^{-\sqrt{3} \frac{h}{\theta}} \]
  - Matern 5/2
    \[ k|h| = |x - y| = \left(1 + \sqrt{5} \frac{h}{\theta} + \frac{5}{3} \left(\frac{h}{\theta}\right)^2\right) e^{-\sqrt{5} \frac{h}{\theta}} \]
Focus on stochastic optimization

Related [R] packages

**DiceKriging**

- Ordinary / Universal kriging
- Nugget / Noise
- Kernel: \( C(x, y) = s^2 k(x, y) \)
  - Exponential
    \[ k(h = |x - y|) = e^{-\frac{h}{\theta}} \]
  - Gauss
    \[ k(h = |x - y|) = e^{-\left(\frac{h}{\theta}\right)^2} \]
  - Power-exponential
    \[ k(h = |x - y|) = e^{-\left(\frac{h}{\theta}\right)^p} \]
  - Matern 3/2
    \[ k(h = |x - y|) = \left(1 + \sqrt{3} \frac{h}{\theta}\right) e^{-\sqrt{3}\frac{h}{\theta}} \]
  - Matern 5/2
    \[ k(h = |x - y|) = \left(1 + \sqrt{5} \frac{h}{\theta} + \frac{5}{3} \left(\frac{h}{\theta}\right)^2\right) e^{-\sqrt{5}\frac{h}{\theta}} \]
Focus on stochastic optimization

Related [R] packages

DiceOptim

- Expected Improvement (local) criterion
  - Low cost criterion (compared to global ones)
  - Maybe extended to noisy kriging
- Parallel maximization of EI based on « Constant Liar » heuristic
  - min/max/mean/kriging mean/…
  - Used as a tuning parameter for global/local optimization
- Integrated / Decoupled call of cost function
  - Suitable for « computer experiments » framework
Focus on stochastic optimization

Related [R] packages

**DiceView**

d <- 2; n <- 16
design.fact <- expand.grid(seq(0, 1, length = 4), seq(0, 1, length = 4))
design.fact <- data.frame(design.fact); names(design.fact) <- c("x1", "x2")
y <- branin(design.fact)
m <- km(design = design.fact, response = y)

sectionview(m, center = c(.3, .3))

sectionview3d(m)
Focus on stochastic optimization

- Integration of stochastic simulator

**Controlled convergence heuristic for MC criticality code**
- MC code => sd estimator of Gaussian output (k-effective) is available (each MC sample increase, for instance)
- Code endpoint may be
  - Sample size :
  - Estimator sd target
  - Estimator quantile (0.999) target
- Early endpoint reached when k-effective 99.9%-quantile << 1.0
  OR
- Early endpoint reached when k-effective 99.9%-quantile << current max.

**Resource-constraint optimization using simulator fidelity**
- On-line control of experiments fidelity using « Start & Stop »
- Quantile based Expected Improvement
- *To be published in Technometrics 2011*
Promethee workbench [http://www.irsn.fr/promethee](http://www.irsn.fr/promethee)

Dice* kriging packages ([http://dice.emse.fr/](http://dice.emse.fr/))

DiceKriging
[http://cran.r-project.org/web/packages/DiceKriging/index.html](http://cran.r-project.org/web/packages/DiceKriging/index.html)

DiceOptim
[http://cran.r-project.org/web/packages/DiceOptim/index.html](http://cran.r-project.org/web/packages/DiceOptim/index.html)

DiceView
[http://cran.r-project.org/web/packages/DiceView/index.html](http://cran.r-project.org/web/packages/DiceView/index.html)