Emulating expensive decision choices with application to computer models of complex physical systems

Jonathan A Cumming and Michael Goldstein

July 2012
General Problem Description
We are considering problems with the following general structure:

- We are interested in some complex system, $y$
- We want to make a decision, $\delta$, that may or may not affect the system
- We enact that decision, and take observations of the system $z$
- Using what we learn from $z$, we make a final decision and take action $\alpha$
- Depending on $\delta$, $\alpha$, and the system $y$, we receive some ultimate reward expressed as a utility
- We want to find good choices of $\delta$
Our First Decision and the System

- In the real world, we have a complex system.
- Denote the state of the complex system as $\theta$, and the portion of the system that is observable (and interesting) as $y$. 
In the real world, we have a complex system.

Denote the state of the complex system as $\theta$, and the portion of the system that is observable (and interesting) as $y$.

We make the decision $\delta$ and do it.
Our First Decision and the System

- In the real world, we have a complex system.
- Denote the state of the complex system as $\theta$, and the portion of the system that is observable (and interesting) as $y$.
- We make the decision $\delta$ and do it.
- The decision $\delta$ may be active, e.g. an intervention, and so may change the state of $y$. 
In the real world, we have a complex system. Denote the state of the complex system as \( \theta \), and the portion of the system that is observable (and interesting) as \( y \). We make the decision \( \delta \) and do it. The decision \( \delta \) may be active, e.g. an intervention, and so may change the state of \( y \). Alternatively, \( \delta \) could be passive, e.g. an observation, which would not affect \( y \) but would affect how and what we see of \( y \), via \( z \).
In the real world, we have a complex system.

Denote the state of the complex system as $\theta$, and the portion of the system that is observable (and interesting) as $y$.

We make the decision $\delta$ and do it.

The decision $\delta$ may be active, e.g. an intervention, and so may change the state of $y$.

Alternatively, $\delta$ could be passive, e.g. an observation, which would not affect $y$ but would affect how and what we see of $y$, via $z$.

Additionally, our decision may not be perfectly executed so there could be some uncertainty on what action was really taken in the world.
Having made our decision $\delta$, we observe the (possibly changed) system, $z$
Our Second Decision

- Having made our decision $\delta$, we observe the (possibly changed) system, $z$
- Depending what we learn from the observations, we take the action $\alpha$
Our Second Decision

- Having made our decision $\delta$, we observe the (possibly changed) system, $z$
- Depending what we learn from the observations, we take the action $\alpha$
- Action $\alpha$ will give us utility $U(\delta, \alpha, \theta)$ – a combination of the costs of performing the decisions we chose, and the rewards for doing so
Our Second Decision

- Having made our decision $\delta$, we observe the (possibly changed) system, $z$
- Depending what we learn from the observations, we take the action $\alpha$
- Action $\alpha$ will give us utility $U(\delta, \alpha, \theta)$ – a combination of the costs of performing the decisions we chose, and the rewards for doing so
- $U(\delta, \alpha, \theta)$ depends on the actions we have taken and the true state of the system
Decisions, Decisions

- The first decision, $\delta$, is directly controllable
Decisions, Decisions

- The first decision, $\delta$, is directly **controllable**
- The second decision, $\alpha$, is determined by **what we learn of the unknown state of the system**
Decisions, Decisions

- The first decision, $\delta$, is directly controllable
- The second decision, $\alpha$, is determined by what we learn of the unknown state of the system
- In the real world, we will only get to do this once
Decisions, Decisions

- The first decision, $\delta$, is directly controllable.
- The second decision, $\alpha$, is determined by what we learn of the unknown state of the system.
- In the real world, we will only get to do this once.
- So what is a good choice for $\delta$?
Decisions, Decisions

- The first decision, $\delta$, is directly **controllable**
- The second decision, $\alpha$, is determined by **what we learn of the unknown state of the system**
- In the real world, we will only get to do this **once**
- So what is a good choice for $\delta$?
- To answer this, we will use a **computer model for $y$**
Enter the simulator

- The computer model $f$ will mimic the behaviour of $y$
Enter the simulator

- The computer model $f$ will mimic the behaviour of $y$, and by adding some noise $z$
Enter the simulator

- The computer model $f$ will mimic the behaviour of $y$, and by adding some noise $z$
- The simulator takes two sets of parameters
  - $x$, the usual input parameters
Enter the simulator

- The computer model $f$ will mimic the behaviour of $y$, and by adding some noise $z$
- The simulator takes two sets of parameters
  1. $x$, the usual input parameters
  2. $d$, the decision as enacted in the model
Enter the simulator

- The computer model $f$ will mimic the behaviour of $y$, and by adding some noise $z$
- The simulator takes two sets of parameters
  1. $x$, the usual input parameters
  2. $d$, the decision as enacted in the model
- We have obvious connections to the system state $\theta$ and the chosen decision $\delta$ in the real world
Emulation

- We can explore the behaviour of $f$ over $d$ and $x$ by making many evaluations of $d$ and $x$. 
Emulation

- We can explore the behaviour of $f$ over $d$ and $x$ by making many evaluations of $d$ and $x$ – but learning about $f$ isn’t really the goal
Emulation

- We can explore the behaviour of $f$ over $d$ and $x$ by making many evaluations of $d$ and $x$ – but learning about $f$ isn’t really the goal.
- Ideally, we want to propagate our model runs through the decision process.
Emulation

- We can explore the behaviour of $f$ over $d$ and $x$ by making many evaluations of $d$ and $x$ – but learning about $f$ isn’t really the goal.
- Ideally, we want to propagate our model runs through the decision process.
- So we need many evaluations over $d$ and $x$ and a mechanism for pushing them through the decision process.
- At the end, we get utilities which are a function of $d$ and $\theta$. 
Emulation

- We can explore the behaviour of $f$ over $d$ and $x$ by making many evaluations of $d$ and $x$ – but learning about $f$ isn’t really the goal.
- Ideally, we want to propagate our model runs through the decision process.
- So we need many evaluations over $d$ and $x$ and a mechanism for pushing them through the decision process.
- At the end, we get utilities which are a function of $d$ and $\theta$.
- So we emulate and optimise $\mathcal{U}(\delta)$. 
The problem as a decision tree

Decision

Observation

Action

System

$U(\delta, \alpha, \theta)$
The decision problem has shifted the focus of the analysis from $f$ to $\mathcal{U}$. 

Broad objectives

1. Construct a design over $(x, d)$ space
2. Evaluate the simulator $f(x, d)$ at every point in the design
3. For every $z(x, d)$, run of the simulator into actions
4. For each run, evaluate the expected utility value, $U(\delta, \alpha, \theta)$, for each run
5. Emulate $U(\delta)$ over the decision space as the 'output' of the model
Broad objectives

- The decision problem has shifted the focus of the analysis from $f$ to $\mathcal{U}$
- Our general approach to this type of problem will be as follows:
Broad objectives

- The decision problem has shifted the focus of the analysis from $f$ to $u$
- Our general approach to this type of problem will be as follows:
  1. Construct a design over $(x, d)$ space
Broad objectives

- The decision problem has shifted the focus of the analysis from $f$ to $\mathcal{U}$
- Our general approach to this type of problem will be as follows:
  1. Construct a design over $(x, d)$ space
  2. Evaluate the simulator $f(x, d)$ at every point in the design
The decision problem has shifted the focus of the analysis from $f$ to $\mathcal{U}$

Our general approach to this type of problem will be as follows:
1. Construct a design over $(x, d)$ space
2. Evaluate the simulator $f(x, d)$ at every point in the design
3. For every $z(x, d)$, run of the simulator into actions
**Broad objectives**

- The decision problem has shifted the focus of the analysis from $f$ to $\mathcal{U}$
- Our general approach to this type of problem will be as follows:
  1. Construct a design over $(x, d)$ space
  2. Evaluate the simulator $f(x, d)$ at every point in the design
  3. For every $z(x, d)$, run of the simulator into actions
  4. For each run, evaluate the expected utility value, $\mathcal{U}(\delta, \alpha, \theta)$, for each run

Steps 1, 2, and 5 are standard emulation
Steps 3 and 4 are different and introduced by the decision analysis

In order to apply this methodology we need to be able to transform each of our runs into actions
Broad objectives

- The decision problem has shifted the focus of the analysis from $f$ to $U$
- Our general approach to this type of problem will be as follows:
  1. Construct a design over $(x, d)$ space
  2. Evaluate the simulator $f(x, d)$ at every point in the design
  3. For every $z(x, d)$, run of the simulator into actions
  4. For each run, evaluate the expected utility value, $U(\delta, \alpha, \theta)$, for each run
  5. Emulate $U(\delta)$ over the decision space as the ‘output’ of the model
The decision problem has shifted the focus of the analysis from $f$ to $\mathcal{U}$.

Our general approach to this type of problem will be as follows:

1. Construct a design over $(x, d)$ space
2. Evaluate the simulator $f(x, d)$ at every point in the design
3. For every $z(x, d)$, run of the simulator into actions
4. For each run, evaluate the expected utility value, $\mathcal{U}(\delta, \alpha, \theta)$, for each run
5. Emulate $\mathcal{U}(\delta)$ over the decision space as the ‘output’ of the model

Steps 1, 2, and 5 are standard emulation
Broad objectives

- The decision problem has shifted the focus of the analysis from $f$ to $U$
- Our general approach to this type of problem will be as follows:
  1. Construct a design over $(x, d)$ space
  2. Evaluate the simulator $f(x, d)$ at every point in the design
  3. For every $z(x, d)$, run of the simulator into actions
  4. For each run, evaluate the expected utility value, $U(\delta, \alpha, \theta)$, for each run
  5. Emulate $U(\delta)$ over the decision space as the ‘output’ of the model
- Steps 1, 2, and 5 are standard emulation
- Steps 3 and 4 are different and introduced by the decision analysis
The decision problem has shifted the focus of the analysis from $f$ to $\mathcal{U}$.

Our general approach to this type of problem will be as follows:

1. Construct a design over $(x, d)$ space
2. Evaluate the simulator $f(x, d)$ at every point in the design
3. For every $z(x, d)$, run of the simulator into actions
4. For each run, evaluate the expected utility value, $\mathcal{U}(\delta, \alpha, \theta)$, for each run
5. Emulate $\mathcal{U}(\delta)$ over the decision space as the ‘output’ of the model

Steps 1, 2, and 5 are standard emulation.

Steps 3 and 4 are different and introduced by the decision analysis.

In order to apply this methodology we need to be able to transform each of our runs into actions.