

SUPPORTING TIME-CRITICAL DECISION MAKING WITH REAL TIME SIMULATIONS

Russell Cheng

School of Mathematics
University of Southampton
Southampton, SO17 1BJ, UNITED KINGDOM

ABSTRACT

This paper describes the use of real time simulation to aid time-critical decision making. An example of such a situation is the provision of a fire and rescue service response to an emergency. Another example is in a battle situation where a field commander has to take a rapid decision on how best to deploy troops. If a simulation model is available that can be run sufficiently fast this can be used to evaluate the likely outcome of different possible decisions before the real decision is actually made, and so provide information on the likely consequences. The methodology of using a simulation model in this way is discussed and applied to an example from the fire and rescue service.

1 INTRODUCTION

This paper describes the use of real time simulation as an aid to time-critical decision making. There are many situations where a computer simulation model can be used in this way. In the WSC 2011 session: Real Time Decision Support, Mousavi, Komashie and Tavakoli (2011) give case studies arising from healthcare and manufacturing and Huang et al. (2011) describe use of simulation to estimate system performance when managing surface transport systems.

Many time-critical decision making problems reduce to being an optimization problem where the objective is to find a good way of operating the system. One version of the problem is where the system performance can be regarded as being a smooth function of a number of continuously varying decision variables and the problem is to find those values of these decision variables that will optimize system performance. This is essentially the *classical optimization* problem. Another version is where the system can be operated in a finite, but large, number of distinctly separate ways and the problem is to choose a good way, possibly the best, of operating the system. In this latter version of the problem the number of distinct ways can be combinatorially large, when we have what is known as a *combinatorial optimization* problem. The distinction between the two types of problem can become blurred in particular instances.

In a series of papers Cheng (2007, 2008, 2010) discusses both types of problem, but where it is assumed that a simulation program can be used to analyse the different ways of operating the system. In all three papers Cheng considers the case where the different ways to be compared are *selected at random*. Simulation runs are then made to see how the system performs under each, to see which is best. If simulation runs can be carried out sufficiently quickly then this random search optimization (RSO) becomes a straightforward and viable way of using simulation in real time-critical decision making. In notation which will be fully defined in Section 2 we will use $W_{(1)}$ to denote the best solution found amongst those randomly selected for consideration; however it will be convenient to use this notation immediately.

There are two main aspects of RSO that are of interest: (i) the experimental procedure used in carrying out the random search itself leading to the identification of $W_{(1)}$, and (ii) establishing how good the system performance actually is corresponding to this solution is, typically involving calculation of a confidence interval for this quantity. The papers by Cheng cited above concentrate on (i), focusing on how best to balance between the number of points searched and the simulation effort used at each point. Cheng

also discusses (ii), but uses mainly computer intensive methods for calculating quantities like confidence intervals. In the real time decision making context, there will not usually be sufficient time to carry out such computer intensive methods in the time period within which the time-critical decision has to be made, so such an evaluation will usually only be of theoretical interest, carried out off-line, and not part of the time-critical decision making.

The overall purpose of this paper is to consider a practical method which handles aspect (ii) as well as aspect (i). We shall consider a method particularly suited for combinatorial optimization problems and illustrate its use by revisiting an example first considered in Cheng (2007). This example – the so called ‘cover-moves’ problem – is a genuine practical problem that occurs in the fire rescue service.

In Cheng (2008) it is pointed out that this problem can be analysed as a combinatorial optimization problem, though the method used in that paper is more general purpose. In Cheng (2010) a simpler RSO method than that discussed in Cheng (2008) is proposed for handling combinatorial optimization problems, but the behaviour of $W_{(1)}$ is examined by computer intensive methods which are not well suited for time-critical decision problems.

In this paper our starting point is the method proposed in Cheng (2007) and in Cheng (2010). This uses RSO to find $W_{(1)}$. However when it comes to studying the properties of $W_{(1)}$ we do not use computer intensive methods in this paper, but instead used a modified form of the analysis discussed in Cheng (2007) to calculate quantities of practical interest like confidence intervals, doing this sufficiently fast so that they can be done in real time in time-critical applications. We apply our proposed method of analysis to the cover-moves fire rescue example which is very time-critical.

In the next section we describe optimization problems and RSO in general terms whilst in Section 3 we discuss a specific normal model that seems suitable for certain combinatorial optimization problems. In Section 4 we show how to fit this normal model to results obtained by RSO. In Section 5 we describe the cover-moves problem encountered by the fire rescue service when dealing with large incidents, and in Section 6 carry out the analysis of output from an RSO in this problem. A summary is provided in Section 7.

2 OPTIMIZATION AND RANDOM SEARCH

2.1 Classical and Combinatorial Optimization

In this first subsection we consider just the deterministic case. Let $J(\theta)$ be the objective function, typically a system performance measure, that is a continuous function of a vector θ of d continuous decision variables, where θ can be selected from a compact region Θ of R^d . Suppose that we are interested in finding that value of θ which optimizes $J(\theta)$; for simplicity we will suppose that the performance measure is a cost so that we wish to minimize its value. We write the minimum expected performance as

$$\delta = \min_{\theta \in \Theta} J(\theta) \quad (1)$$

and for simplicity we shall only consider the case where this is obtained at an interior point θ_{\min} of Θ . In classical optimization $J(\theta)$ is assumed to have smoothness conditions which allow it to be approximated by a quadratic function of θ in the neighborhood of θ_{\min} . Convergence properties of numerical search procedures are then relatively easy to establish.

In this paper we are interested in certain commonly occurring combinatorial optimization problems. Usually such problems do not obviously fall into the framework of the classical optimization problem just set out, where there is a smooth objective function depending on input variables of given dimension d . However we argue that though a dimension is not usually defined in combinatorial problems, if we were able to do so, then the dimension d would have to be regarded as large. In such a situation, if results from

classical optimization theory were to be at all applicable, then it would have to be for the case when d is assumed large.

We admit that this viewpoint of combinatorial optimization problems is somewhat speculative at this juncture. Thus the main objective of this paper is not to try develop rigorous theory to support our approach, but merely to carry out exploratory examination of a genuinely practical problem, to see how successful such an approach might be. We leave as an open question whether a dimensionality can be associated with combinatorial problems. However irrespective of this, our numerical results do furnish some indication of how high dimension might be treated in numerical optimization.

2.2 Random Search Optimization

Random search can be thought of as being a simple way of exploring the region Θ , supposing that this region does contain θ_{\min} . Note that this immediately injects a probabilistic element into the problem, even if our problem had been initially deterministic. Our starting point is the statistical model of RSO considered by Cheng (2008) in which simulation runs are made to explore the behaviour of $J(\theta)$. We begin by assuming that the random search is carried out as follows. This will not be the final version of the search that we will be discussing in more detail, but serves as a good starting point which allows us to set out the issues.

First sample m mutually independent values of θ :

$$\theta_1, \theta_2, \dots, \theta_m, \quad (2)$$

which we shall call *search points*, from some continuous distribution with density $g(\theta)$, $\theta \in \Theta$. We allow for a general density rather than sampling from a uniform distribution to enable sampling to be focussed in the most promising regions of Θ based on prior information.

Then for each θ_i we make n simulation runs, each of some predetermined and fixed standard length t . We shall not discuss how individual runs are conducted. For example, if a warm up period is needed in each run, we assume that this has already been considered and dealt with. If the total time available allows a maximum of c simulation runs, we have that

$$c = nm.$$

It will be convenient to write the performance measure as $J(\theta) = \mu + X(\theta)$ where μ is a constant to be defined and $X(\theta)$ is the part depending on the decision quantities θ . We write J_i for $J(\theta_i)$ and X_i for $X(\theta_i)$. The important point to note is that because the θ_i are randomly sampled, the J_i and X_i are *also* random variables. In all that follows we shall use the notation $J(\theta)$ and $X(\theta)$ when we are regarding J and X as deterministic functions of θ , and the notation J_i , and X_i , when regarding J and X as a random variables resulting from the random sampling of the θ_i in (2). The constant μ can be viewed as the mean of the random variables J_i , and the X_i as having mean $E(X_i) = 0$.

An additional random variation has to be allowed for when simulation is used to assess system performance. This is because each simulation run outputs a performance measure $J(\theta)$ that is observed with error, even when θ is being regarded as fixed. We assume that the *observed* performance measures, obtained from simulation runs, take the form

$$Y_{ij} = \mu + X_i + \eta_j + \varepsilon_{ij}, \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n. \quad (3)$$

Here we distinguish between two random ‘error’ quantities appearing in a simulation run, even when θ is fixed. The quantities ε_{ij} are assumed to be all mutually independent random errors arising from random numbers sampled independently at all the different design points and replicates. We assume the ε_{ij} have mean zero and variance σ^2 . The quantities η_j are random errors which arise from the use of common numbers. Let \mathbf{z}_j be a set of random numbers used in the j th run made at a given design point, and with this *same* set used in the j th run made at all m design points, i.e. the set \mathbf{z}_j comprises random numbers *common* to all the j th runs across all design points. They therefore give rise to the same random error η_j in the simulation output irrespective of the design point θ_i .

The averages of the observations at each θ_i are

$$W_i = n^{-1} \sum_{j=1}^n Y_{ij} = \mu + X_i + n^{-1} \sum_{j=1}^n \eta_j + n^{-1} \sum_{j=1}^n \varepsilon_{ij} = \mu + X_i + \bar{\eta} + \zeta_i, \text{ say, for } i = 1, 2, \dots, m, \quad (4)$$

where the averaged errors ζ_i have mean zero and variance:

$$\text{Var}[\zeta_i] = \sigma^2 / n = \sigma_n^2, \text{ say.} \quad (5)$$

We include the suffix n as a reminder that $\text{Var}[\zeta_i]$ depends on n .

The effect of using random numbers is to add the *same* random error, $\bar{\eta}$, to all the W_i . This shows the benefit of using random numbers when we compare different W_i , as the $\bar{\eta}$ random error then cancels out.

Equation (4) shows that there are therefore two sources of variation in RSO: the *search induced variability* of the X_i and the *simulation induced variability* of the ζ_i , which is independent of that of the X_i . We shall write $F_X(\cdot)$, and $F_\zeta(\cdot)$ to denote the cumulative distribution functions (CDF) of X_i and ζ and $f_X(\cdot)$ and $f_\zeta(\cdot)$ for their probability density functions (PDF).

It will be convenient to assume that the observations W are put in ranked order

$$W_{(1)} < W_{(2)} < \dots < W_{(m)}.$$

It is natural to take the decision $\theta_{(1)}$ corresponding to $W_{(1)}$, the minimum observed value of the W_i , as the best available of those points searched. Note that, because

$$W_{(1)} = \mu + X_{(1)} + \bar{\eta} + \zeta_{(1)} \quad (6)$$

involves a random error $\zeta_{(1)}$, it may be that $X_{(1)}$, the *actual performance achieved* corresponding to $W_{(1)}$ is not the best amongst the decisions examined. As $X_{(1)}$ is never directly observable it is therefore of particular interest to estimate its distribution. One of the main objectives of this paper is to show how this can be done.

Moreover it is of particular interest to see if this can be done sufficiently quickly so that it can be obtained in time-critical decision taking. A statistical model that allows one to do this is of special interest. In the next section we consider such a model.

3 THE NORMAL MODEL

3.1 RSO when Search Points have High Dimension

Under the classical assumption that $X(\theta)$ is a quadratic function of θ near θ_{\min} it is known, see Chia and Glynn (2007), Cheng (2013), that $W_{(1)}$ has minimum variance when

$$m \sim r c^{d/(d+4)}, \quad n \sim r^{-1} c^{4/(d+4)} \quad (7)$$

as $c \rightarrow \infty$, with r an arbitrary but fixed positive constant.

An immediate consequence of this result is that as $d \rightarrow \infty$, less and less effort is spent making runs at individual design points θ_i , and correspondingly more effort on increasing the number of search points m . If we take $r = 1$ it is clear therefore that for large d we are better off not replicating runs at individual θ_i but should simply take $n = 1$. The only reason for replicating at a given θ_i is to enable estimation of σ^2 by calculating this variance from the replicated observations at each given θ_i .

Once the observations (4) are obtained, our approach is to fit a statistical model to them. To specify a statistical model for the observations (4) we note that W_i is simply the sum three terms: $(\mu + \bar{\eta})$ which is effectively a constant, X_i and ξ_i . Thus all we need to do is specify $F_X(\cdot)$ the distribution of X and $F_\xi(\cdot)$ the distribution of ξ . The distribution of W is then the convolution of these two distributions plus the constant. For the situation where d is large we consider the case where we do not have to specify a definite value for d but can assume that a central limit theorem applies to the distribution of X so that it is asymptotically normally distributed. Moreover we shall only consider the situation where the errors are also normally distributed. The distribution of W is then also normally distributed.

Cheng (2010) discusses the situation when $d \rightarrow \infty$ and shows that it is reasonable to approximate the distribution of X_i by the normal, at least in the left tail. In combinatorial problems we suggest that it may be reasonable to assume that the entire distribution of X_i is adequately represented by a normal distribution. It is beyond the scope of this paper to investigate the precise conditions when this might hold rigorously. We only offer the following rough and heuristic argument when this is might occur.

A combinatorial problem can be regarded as a deterministic situation where the decision space comprises a finite but very large number of choices which we can represent as $\Theta = \{\theta_i \mid i = 1, 2, \dots, N\}$. Suppose that each choice θ_i depends on a large number, d , of factors so that $\theta_i = \theta_i(\theta_i^{(1)}, \theta_i^{(2)}, \dots, \theta_i^{(d)})$ where $\theta_i^{(j)}$ is the value of the j th factor in the i th decision. In RSO we select θ_i at random so that the values of all the factors $\theta_i^{(j)}$, $j = 1, 2, \dots, d$ are selected at random. The factors $\theta^{(j)}$ therefore behave like random variables with $\theta_i^{(j)}$ being the sampled value corresponding to the decision θ_i . It is perfectly possible for the combinations of factor values appearing in the set of choices comprising Θ to be so distributed that, in selecting θ_i at random, we are in effect carrying out independent sampling of the each of the factors $\theta^{(j)}$. Suppose therefore $X_i = X(\theta_i)$ depends on the factors additively so that $X(\theta_i) = \mu + \theta_i^{(1)} + \theta_i^{(2)} + \dots + \theta_i^{(d)}$ where μ is some constant; if the $\theta^{(j)}$, treated as random variables and standardized so that their variances satisfy the Lindeberg condition, see Billingsley (1979) for example, then RSO would produce a set of X_i which would be asymptotically normally distributed as $d \rightarrow \infty$.

For the rest of the paper we consider analysis of the behaviour of $W_{m,n}$ under the assumption that the distribution of X_i is normal, i.e.

$$X_i \sim N(0, \omega^2), \quad (8)$$

where ω is unknown and has to be estimated. It might appear strange that this representation does not involve x_{\min} . But this arises only in the limit as $d \rightarrow \infty$. A practical interpretation is that, as $d \rightarrow \infty$, the X_i , in tending to normality, have a lower limit, i.e. x_{\min} , that is increasingly unstable and unreliable to estimate. Cheng (2010) discusses estimation of a low quantile of (8) instead. In this paper we do not consider this further.

Our statistical model of the observations (3) or (4) is completed by specifying the distribution of ε_{ij} , the random simulation induced error arising from the within-run stochastic variation. We shall simply assume that this is also normal, i.e.

$$\varepsilon \sim N(0, \sigma^2) \quad (9)$$

where σ is unknown but constant.

In summary, for the rest of the paper we make the following assumption.

Assumption A:

- (i) The RSO is for a problem where the dimension of the search points θ_i is large.
- (ii) The *observed* performance measures take the form

$$Y_{ij} = \mu + X_i + \eta_j + \varepsilon_{ij}, \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n. \quad (10)$$

with $X_i \sim N(0, \omega^2)$ and $\varepsilon_{ij} \sim N(0, \sigma^2)$ mutually independent so that (conditional on $\bar{\eta}$) W_i as defined in (4) takes the form

$$W_i = \mu + \bar{\eta} + X_i + \zeta_i \sim N(\mu + \bar{\eta}, \psi^2) \quad (11)$$

where

$$\psi^2 = \omega^2 + n^{-1}\sigma^2. \quad (12)$$

3.2 Estimating $X_{(1)}$

As already discussed, the quantity of most interest in RSO is $X_{(1)}$, the unobserved actual performance achieved when $W_{(1)}$, as given in (6) is selected as being the best search point. The next best thing to obtaining $X_{(1)}$ itself is to estimate its distribution. Under Assumption A, when X_i and ζ_i are both normally distributed and independent, we can calculate $F_{X_{(1)}}(\cdot)$ as follows.

As μ is a constant in (10) and $\bar{\eta}$ is effectively so, we need only focus on $Z_i = X_i + \zeta_i$ the random part of W_i , and consider the probability that $Z_{(1)}$, the smallest Z_i , will occur at each given i and the way that this will occur. Thus suppose that $X_{(1)} = X_i$, for a given i . An elementary conditional argument shows that

$$\Pr(X_{(1)} = X_i \text{ and } X_i > x) = \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\bar{z}-x} f_X(z-u) f_{\zeta}(u) [1 - F_Z(z)]^{m-1} du \right\} dz.$$

Hence, as any of the i are equally likely to be selected, we have

$$\Pr(X_{(1)} < x) = F_{X_{(1)}}(x) = 1 - m \left(\int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\bar{z}-x} f_X(z-u) f_{\zeta}(u) [1 - F_Z(z)]^{m-1} du \right\} dz \right), \quad (13)$$

and

$$f_{X_{(1)}}(x) = m \left(\int_{-\infty}^{\infty} f_X(x) f_{\zeta}(z-x) [1 - F_Z(z)]^{m-1} dz \right). \quad (14)$$

Under Assumption A the distributions used in (13) and (14) are all normal, depending only on the variance parameters ω and σ/\sqrt{n} . Once they have been estimated, estimates of the CDF and PDF of $X_{(1)}$ can be calculated from (13) and (14) by numerical quadrature. In the next subsection we discuss how to estimate the parameters μ , ω and σ under Assumption A.

4 FITTING THE NORMAL MODEL

4.1 Estimating μ , ω^2 and σ^2

We first consider a basic procedure for estimating the parameters. We suppose that the RSO observations take the form (3). We write Y_{ij} and X_i in lower case as a reminder that in the actual observations they are variate values.

We can estimate the x_i and η_j as fixed effects by minimizing the sum of squares

$$S = \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \mu - x_i - \eta_j)^2 \quad (15)$$

subject to

$$\sum_{i=1}^m x_i = \sum_{j=1}^n \eta_j = 0. \quad (16)$$

This gives the estimates

$$\hat{\mu} = (nm)^{-1} \sum_{i=1}^m \sum_{j=1}^n y_{ij}, \quad (17)$$

$$\hat{x}_i = n^{-1} \sum_{j=1}^n y_{ij} - \hat{\mu}, \quad i = 1, 2, \dots, m, \quad (18)$$

$$\hat{\eta}_j = m^{-1} \sum_{i=1}^m y_{ij} - \hat{\mu}, \quad j = 1, 2, \dots, n. \quad (19)$$

The estimate for $\text{Var}[X_i] = \omega^2$ is then:

$$\hat{\omega}^2 = m^{-1} \sum_{i=1}^m \hat{x}_i^2, \quad (20)$$

and estimate of the variance of $\text{Var}[\varepsilon_{ij}] = \sigma^2$ is

$$\hat{\sigma}^2 = (nm)^{-1} \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \hat{\mu} - \hat{x}_i - \hat{\eta}_j)^2. \quad (21)$$

4.2 A Practical RSO Procedure

The basic RSO procedure with observations taking the form (4) is not very satisfactory as we want to have n as small as possible, i.e. to have $n = 1$. However we cannot then estimate σ and consequently will not be able to estimate the distribution of $X_{(1)}$ from (13) or (14). Our suggested practical procedure is to carry out the RSO in two steps.

Step 1: The purpose of this step is to obtain an estimate of σ^2 . For a selected number, say $m = m_1$ of initial search points, carry out RSO with $n > 1$, say $n = n_1 = 5$. The parameter σ is then estimated from (21).

Step 2: Carry out the RSO proper, using m_2 search points, but now with $n = n_2 = 1$. The parameter ω is then estimated from equations (17)-(20), with $m = m_2$ and $n = 1$.

5 AN EXAMPLE

This example is taken from Cheng (2007). To save the reader having to access that paper for details, we summarize its main features here. The example is drawn from a very specific genuine application based on work done for the fire service emergency cover (FSEC) Section of UK Government Department of Communities and Local Government (DCLG).

5.1 Fire Service Emergency Cover

Regional Fire Brigades in the UK possess a very sophisticated tool for gathering and analysing incident data in a very comprehensive way. This information is used for planning and to provide operational statistics to the UK Government. It is realised by brigade management that the data could be used to inform day to day management decisions.

The specific question of interest was whether it would be possible to develop a simulation model that would run sufficiently fast to be used to evaluate risk in real time. Such a model might then be deployed as an operational tool to provide real-time advice to brigade officers in responding to actual incidents.

The speed at which such a model can be run is a determining factor of its practicality. In our case the discrete event simulation (DES) model, which was a very detailed one, was able to simulate a year's operation in about three seconds.

5.2 The 'Cover-Moves' Problem

An operational problem of particular interest is the *Cover-Moves Problem*. This occurs when a fire brigade responds to a large incident (one that needs a large, say 8 or more, number of fire appliances to attend). The incident controller then usually repositions a small number of vehicles not involved in the large incident in what are called *cover-moves*, to try to minimize risk in the remainder of the region. Here risk can be clearly defined. We shall take as our performance measure the expected fatality rate (as measured by the expected number of fatalities over a given period of time, under the conditions of the large incident). For simplicity in what follows, we refer to this as the (*observed*) *fatality count*.

The choice of a worthwhile cover-move combination (CMC) is an example of a problem in combinatorial optimization. It is not usually possible, certainly in real time, to identify the best solution. The real question is whether a worthwhile operational solution can be found.

We consider the kind of cover-moves solution achievable. In one example of a typical large incident, consideration was given to selecting 3 vehicles for cover-moves out of 16 available vehicles located in 11 stations. The 3 vehicles were to be sent to 3 out of the 6 stations that had supplied vehicles to attend the large incident. A simple combinatorial calculation shows that there are 25800 distinct CMCs possible. In fact, the majority of these could be ruled out on operational grounds so that only 230 CMCs turned out in this instance to need serious consideration.

The strict (policy driven) operating requirement for the cover-moves problem is that a solution has to be found within *one minute* of the notification of occurrence of a large incident.

In our example, if simulations of a year's operation were used, then one could only make 20 runs in the one minute allowed. However if, say, each run simulated only two months' operation, then about 120 runs of the simulation model could be made. As the example is for discussion purposes only, we actually carried out simulation runs for all 230 CMCs each of two months' operation. This provides a benchmark to gauge how well our proposed real-time method would have worked out in this particular example. Figure 1 gives the empirical distribution of the observed fatality rates (i.e. the Y_i) for the 230 CMCs.

Cheng

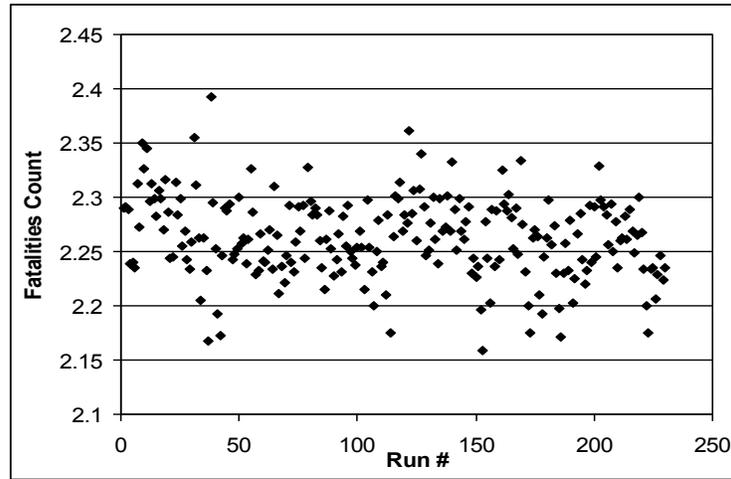


Figure 1: Fatality count by run number for 230 CMCs

A run carried out for the situation when no large incident has occurred gave a fatalities count of 2.108, well below all the cases where the large incident has occurred, even with cover moves. A run where the large incident has occurred but no cover moves are made gave a fatality count of 2.278 which is lower than many of the fatality counts where a CMC was deployed. This shows that it is very possible for a CMC to be counter-productive.

6 REAL-TIME ANALYSIS OF THE COVER MOVES PROBLEM

We apply the method set out in Section 4.2 to analyse the cover moves problem.

In Step 1 of the RSO process we took $m_1 = 5$ and $n = 4$. Figure 2 shows the 20 observed values with the horizontal scale corresponding to the observed fatality counts. The five observations of each replicate are on the same horizontal line, with one line for each replicate. It will be seen that there is some significant variation in horizontal position of each group – this is due to the variation of η_j in (10) –, but that within each group the ranked order of the points remains very similar – thus the ranking is determined largely by X_i in (10), with the effect of ε_{ij} being much smaller in comparison.

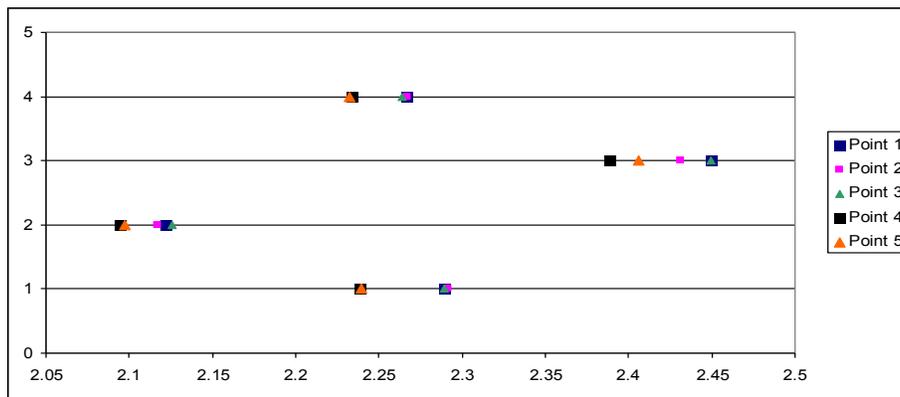


Figure 2. Fatality counts obtained in the 20 simulation runs of Step 1 of the suggested RSO process

These observations were used to estimate σ from equation (21) giving

$$\hat{\sigma} = 0.00622 ,$$

For Step 2 of the RSO process we set $m_2 = 100$, $n = 1$. These observations correspond to the first 100 values depicted in Figure 1. The lowest value in this set is 2.168, the 37th value. Note that the first 100 observations happen not to be a particular good set of values compared with later CMCs, but of course in the time-critical situation where only 100 observations are made, we would not know this. Fitting the RSO model to these 100 observations gave the estimates

$$\hat{\mu} = 2.267, \hat{\omega} = 0.0372 .$$

Using these values allows us to calculate the distribution of $X_{(1)}$ from (13) or (14). Note that we are only using the Step 2 observations so that $Var[\xi_i] = \sigma^2 / n = \sigma^2$ in this case as $n = 1$. We have calculated the fitted PDF numerically from (14) using simple quadrature by Simpson's Rule with appropriately selected ranges of integration over which the integrand is not negligible. Equation (14) only shows how the distribution of $X_{(1)}$ depends on the X_i and ξ_i . These both have zero expectations. We therefore need to add the overall mean $\hat{\mu} = 2.267$ to the argument of the PDF if this is to be the fatality rate. Figure 3 shows the behaviour of the density for the case $m_2 = 100$, corresponding to number of observations used in the Step 2 of the RSO process. The position where the depicted ordinate axis crosses the horizontal fatality count axis has been placed at $x = 2.278$. the value of the fatality count when cover moves are not deployed following despatch of appliances to the large incident. This thus corresponds to the breakeven fatality count with $X_{(1)}$ having to be less than this value to be effective. As one would wish, the bulk of the probability distribution of $X_{(1)}$ is significantly to the left of this value. However the small probability to the right of $x = 2.278$ shows that there is still a noticeable probability that the best value estimated by the RSO is less than effective.

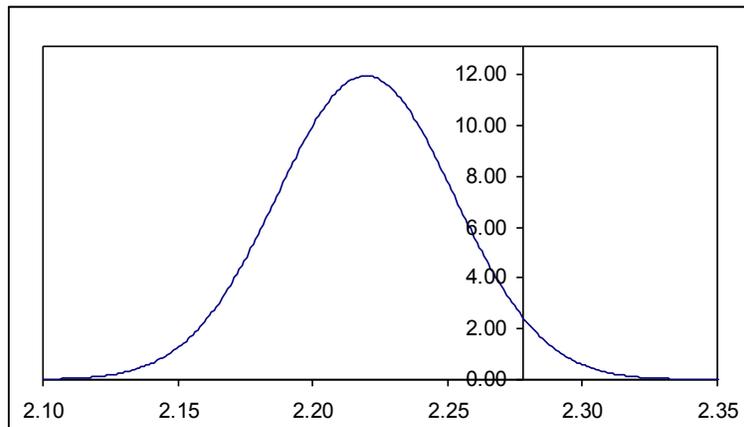


Figure 3: Estimated distribution of $X_{(1)}$ the expected fatality count given by the cover move combination found by RSO under the one minute constraint within which the decision has to be made. The point where the y axis crosses the x -axis corresponds to the fatality count where the large incident has occurred, but no cover moves are deployed

The final figure, Figure 4, shows the EDF of observed fatalities in the $m_2 = 100$ observations of Step 2 of the suggest RSO process together with the CDF of the fitted distribution. This latter is as given by

equations (11) and (12) with $\hat{\mu} = 2.267$, $\hat{\omega} = 0.0372$ and $\hat{\sigma} = 0.00622$. It has to be admitted that though the general fit is not unreasonable, the fit is poorest in the left hand tail and this is likely to have the most influence on the estimated distribution of $X_{(1)}$. However it seems likely that this lack of fit will not have much influence in estimating the probability that the chosen cover move will have *some* beneficial effect. This is clearly governed by those search points, for which there is some beneficial effect. As Figure 4 shows, these make up almost exactly 60% of the search points. The fitted CDF matches this proportion rather accurately at this x -value. We argue therefore that, whatever the true shape of the PDF of $X_{(1)}$, the value of the area under the PDF to the left of $x = 2.278$ in Figure 4 will be close to that under the true PDF to the left of $x = 2.278$.

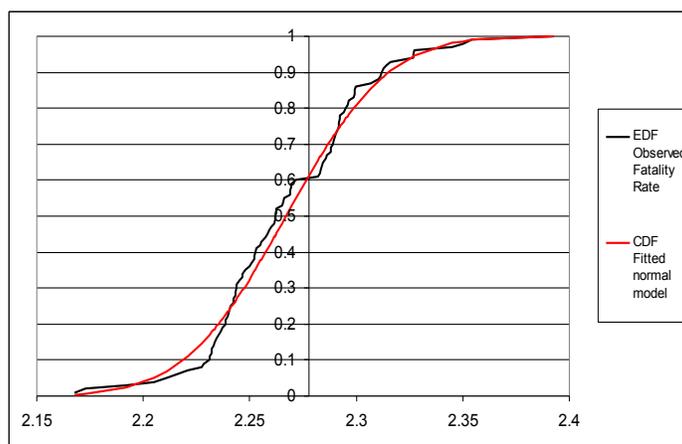


Figure 4: The EDF of the $m_2 = 100$ observations of Step 2 of the suggest RSO process, and the CDF of the fitted distribution. The value where the x -axis is crossed by the y axis has been set so that it corresponds to the fatality count where the large incident has occurred, but no cover moves are deployed

7 CONCLUDING REMARKS

This paper has focused on the use of a particular normal statistical model that allows a relatively easy analysis of the results obtained when one is attempting to improve system performance using random search optimization (RSO). RSO is very easy to implement, and indeed it is little different from the initial exploratory simulation runs carried out in developing a simulation model and in a preliminary evaluation of system performance.

If the simulation model runs fast enough and if it is to be used in improving system performance, then, in situations that involve real time adjustment of the system, the simulation model can be incorporated as a decision tool to assist in this process.

In the real time context, it may be unrealistic to expect true system optimization to be achievable. One should simply be satisfied with obtaining improved system performance. As a bottom line one would like some reassurance that the best way of running the real system found by RSO actually does lead to *some* improvement. This paper shows how to calculate a probability distribution which estimates how the unknown actual true system is likely to behave if it is run according to the best solution found by RSO. This calculation can be done sufficiently fast to be incorporated in the real time decision making, so that an indication is available of how well the best solution found by RSO might actually perform.

It may be thought that the proposed normal model may be too narrowly applicable to be of general practical use. However as illustrated in the numerical example, it may be sufficiently robust to at least give a clear indication of whether the RSO process has been worthwhile or not.

REFERENCES

- Billingsley, P. 1979. *Probability and Measure, 2nd.Edition*. New York: Wiley.
- Cheng, R.C.H. 2007. Determining efficient simulation run lengths for real time decision making In *Proceedings of the 2007 Winter Simulation Conference S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.* IEEE, Piscataway, 340-345.
- Cheng, R.C.H. 2008. Simulation Assisted Optimization by Random Search. In *Proceedings of the 2008 Operational Research Society Simulation Workshop (SW08)*. Eds K. Kotiadis, C.S.M. Currie, S. Robinson and S.J.E. Taylor. OR Society, 79-85.
- Cheng, R.C.H. 2010. Random Search in High Dimensional Stochastic Optimization. In *Proceedings of the 2010 Winter Simulation Conference B. Johansson, S. Jain, J. Montoya-Torres, J. Huan, and E. Yücesan, eds.* IEEE, Piscataway, 1012-1023.
- Cheng, R.C.H. 2013. Fitting Statistical Models of Random Search in Simulation Studies. *ACM Transactions on Modeling and Computer Simulation*. To appear.
- Chia, Y. L. and Glynn, P. W. 2007. Optimal convergence rate for random search. In *Proceedings of the 2007 INFORMS Simulation Society Workshop Eds C-H Chen and S.G. Henderson*
- Huang, Y.-L., Suh, W., Alexopoulos, C., Fujimoto, R. and Hunter, M. 2011. Statistical Issues in Ad-Hoc Simulations. In *Proceedings of the 2011 Winter Simulation Conference S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.* IEEE, Piscataway, 612-624.
- Mousavi, A., Komashie, A. and Tavakoli, S. 2011. Simulation-based Real-Time Performance Monitoring (SIMMON): A Platform for Manufacturing and Healthcare Systems. In *Proceedings of the 2011 Winter Simulation Conference S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.* IEEE, Piscataway, 600-611.

AUTHOR BIOGRAPHY

RUSSELL C. H. CHENG is Emeritus Professor of Operational Research at the University of Southampton. He has an M.A. and the Diploma in Mathematical Statistics from Cambridge University, England. He obtained his Ph.D. from Bath University. He is a former Chairman of the U.K. Simulation Society, a Fellow of the Royal Statistical Society and Fellow of the Institute of Mathematics and Its Applications. His research interests include: design and analysis of simulation experiments and parametric estimation methods. He was a Joint Editor of the *IMA Journal of Management Mathematics*. His email and web addresses are <R.C.H.Cheng@soton.ac.uk> and <www.personal.soton.ac.uk/rchc> .