COMPARISON OF RESPONSE SURFACE METHODOLOGY AND THE NELDER AND MEAD SIMPLEX METHOD FOR OPTIMIZATION IN MICROSIMULATION MODELS

H. Gonda Neddermeijer[a,b,*], Nanda Piersma[b], Gerrit J. van Oortmarssen[a],
J. Dik F. Habbema[a], Rommert Dekker[b]

[a] Department of Public Health, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
[b] Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
[*]Corresponding author. Fax: +31 10 408 9162; e-mail: neddermeijer@few.eur.nl.

ECONOMETRIC INSTITUTE REPORT EI-9924/A

Abstract

Microsimulation models are increasingly used in the evaluation of cancer screening. Latent parameters of such models can be estimated by optimization of the goodness-of-fit. We compared the efficiency and accuracy of the Response Surface Methodology and the Nelder and Mead Simplex Method for optimization of microsimulation models. To this end, we tested several automated versions of both methods on a small microsimulation model, as well as on a standard set of test functions. With respect to accuracy, Response Surface Methodology performed better in case of optimization of the microsimulation model, whereas the results for the test functions were rather variable. The Nelder and Mead Simplex Method performed more efficiently than Response Surface Methodology, both for the microsimulation model and the test functions.

Keywords: Optimization; Simulation; Health

1 Introduction

In this paper we investigate the performance of two classes of algorithms for optimization of stochastic simulation models: algorithms based on Response Surface Methodology and variants of the Nelder and Mead Simplex Method. Our particular interest in these algorithms stems from the need for efficient minimization algorithms that can be used in optimizing microsimulation models of disease control. In microsimulation models individual fictitious life histories, including disease processes and the impact of intervention, are simulated.

For example, cancer screening microsimulation models are used in the evaluation of mass cancer screening programmes. To make inferences about parameters that cannot be observed directly, such as duration of preclinical screendetectable stages of cancer (Day and Walter, 1984), a cancer screening microsimulation model is fitted on observed screening results and cancer incidence data by optimizing the goodness-of-fit of the model (van Oortmarssen et al., 1990). Such a fitting procedure involves optimization of an objective function that can only be observed indirectly from the microsimulation model, which gives a stochastic response function that cannot be given explicitly as a function of the parameters.

Consequently, microsimulation models are often considered as stochastic black-box models (Pflug, 1996), where the optimization routine acts as a shell around the existing microsimulation program and only uses observations of the stochastic response function. In this paper, we consider microsimulation models for which all parameters included in the optimization are real-valued numbers. An optimization algorithm for microsimulation models should be efficient in terms of the number of evaluations needed for finding an optimum, since function evaluations (i.e. runs of the microsimulation model) are computationally expensive. The algorithm should be reliable, in the sense that repeated optimizations should give comparable results. Furthermore, the algorithm should be accurate, in the sense that an observed optimum should be close to the real optimum. Accuracy is required in statistical comparison of different parameterizations of a model where one should be confident that indeed the best fitting models are compared.

Among optimization methods that only use observations of the stochastic response function are the Nelder and Mead Simplex Method, Stochastic Approximation, Response Surface Methodology and Simultaneous Perturbation Stochastic Approximation (see Kleijnen, 1974; Jacobson and Schruben, 1989; Spall, 1992; Fu, 1994). Both the Nelder and Mead Simplex Method and Response Surface Methodology are frequently used for the optimization of simulation models. However, there are surprisingly few papers in which the performances of these optimization methods are compared systematically. Barton and Ivey (1996) investigated the performance of the Nelder and Mead Simplex Method in simulation optimization, and studied various modifications of the method that might improve its performance. Response Surface Methodology sequential procedures provide a very general methodology for optimization via simulation (Fu, 1994). The aim of this paper is to compare the two most successful Nelder and Mead Simplex Method algorithms (according to Barton and Ivey) with several algorithms based on Response Surface Methodology. We test the automated algorithms using a standard set of deterministic test functions for unconstrained optimization and an existing microsimulation model.

The remainder of this paper is organized as follows. In the next section the optimization methods will be described. Next, we will describe how we will test these methods and give the results of the tests. We will conclude with some recommendations.

2 Optimization Methods

We consider the minimization of an objective function $f(\boldsymbol{x}) = E(F(\boldsymbol{x}))$, $\boldsymbol{x} \in D \subset \Re^n, n \geq 1$, where $F(\boldsymbol{x})$ denotes the stochastic response function of the simulation model and $E(F(\boldsymbol{x}))$ denotes its expected value. The arguments $\boldsymbol{x} = (x_1,...,x_n)$ represent the parameters of the microsimulation model. In this section we will describe the Nelder and Mead Simplex Method (NMSM) and Response Surface Methodology (RSM). The issue of selecting a criterion for ending the optimization procedures is not addressed in this study. Therefore, we use for both NMSM and RSM a predefined fixed, large number of evaluations.

2.1 The Nelder and Mead Simplex method

NMSM is a direct-search method that has shown a good performance on both deterministic objective functions (Nelson, 1985) and stochastic functions (Dennis and Woods, 1987). A detailed description of the algorithm can be found in (Barton and Ivey, 1996).

For the minimization of a function of n variables, NMSM defines a *simplex* with (n+1) vertices. During an iteration, the objective function is evaluated at each vertex of the simplex, and the vertex with the lowest value ($\boldsymbol{x}_{low}$), the vertex with the highest value ($\boldsymbol{x}_{hi}$) and the vertex with the next-to-highest value ($\boldsymbol{x}_{nexthi}$) are determined. Vertex $\boldsymbol{x}_{hi}$ is reflected through the centroid $\boldsymbol{x}_0$ of the remaining vertices to find a new vertex ($\boldsymbol{x}_{refl}$):
$\boldsymbol{x}_{refl} = (1+a)\boldsymbol{x}_0 - a\boldsymbol{x}_{hi}, a > 0$, and the objective function is evaluated in vertex $\boldsymbol{x}_{refl}$. Next, a new simplex is constructed as follows:

- If $F(\boldsymbol{x}_{refl}) \geq F(\boldsymbol{x}_{hi})$, then the objective function is evaluated in a contracted vertex between $\boldsymbol{x}_{hi}$ and $\boldsymbol{x}_0$, defined by $\boldsymbol{x}_{contr1} = b\boldsymbol{x}_{hi} + (1-b)\boldsymbol{x}_0$, $0 < b < 1$. If $F(\boldsymbol{x}_{contr1}) < F(\boldsymbol{x}_{hi})$, then the new simplex is found by replacing vertex $\boldsymbol{x}_{hi}$ by vertex $\boldsymbol{x}_{contr1}$, otherwise the new simplex is found by shrinking the current simplex around vertex $\boldsymbol{x}_{low}$, by replacing vertex $\boldsymbol{x}_i$ by $d\boldsymbol{x}_i + (1-d)\boldsymbol{x}_{low,}\, \boldsymbol{x}_i \neq \boldsymbol{x}_{low}$, $0 < d < 1$.

- If $F(\boldsymbol{x}_{nexthi}) < F(\boldsymbol{x}_{refl}) < F(\boldsymbol{x}_{hi})$ then the objective function is evaluated in a contracted vertex between $\boldsymbol{x}_{refl}$ and $\boldsymbol{x}_0$, defined by $\boldsymbol{x}_{contr2} = b\boldsymbol{x}_{refl} + (1-b)\boldsymbol{x}_0$, $0 < b < 1$. If $F(\boldsymbol{x}_{contr2}) < F(\boldsymbol{x}_{refl})$, then the new simplex is found by replacing vertex $\boldsymbol{x}_{hi}$ by vertex $\boldsymbol{x}_{contr2}$, otherwise the new simplex is found by shrinking the current simplex around vertex $\boldsymbol{x}_{low}$, by replacing vertex $\boldsymbol{x}_i$ by $d\boldsymbol{x}_i + (1-d)\boldsymbol{x}_{low,}\, \boldsymbol{x}_i \neq \boldsymbol{x}_{low}$, $0 < d < 1$.

- If $F(\boldsymbol{x}_{low}) \leq F(\boldsymbol{x}_{refl}) \leq F(\boldsymbol{x}_{nexthi})$ then the new simplex is found by replacing vertex $\boldsymbol{x}_{hi}$ by vertex $\boldsymbol{x}_{refl}$.

- If $F(\boldsymbol{x}_{refl}) < F(\boldsymbol{x}_{low})$ then the objective function is evaluated in an expanded vertex between $\boldsymbol{x}_{refl}$ and $\boldsymbol{x}_0$, defined by $\boldsymbol{x}_{\exp} = g\boldsymbol{x}_{refl} + (1-g)\boldsymbol{x}_0$, $g > 1$. If $F(\boldsymbol{x}_{\exp}) < F(\boldsymbol{x}_{low})$, then the new simplex is found by replacing vertex $\boldsymbol{x}_{hi}$ by vertex $\boldsymbol{x}_{\exp}$, otherwise the new simplex is found by replacing vertex $\boldsymbol{x}_{hi}$ by vertex $\boldsymbol{x}_{refl}$.

The next iteration begins with the new simplex. If during an iteration a vertex is defined outside the feasible region *D*, then this vertex is projected onto the boundary of this region. The initial simplex is given by

$$\left\{(x_1^0,...,x_n^0),(x_1^0+c_1,...,x_n^0),...,(x_1^0,...,x_n^0+c_n)\right\}$$

where $\boldsymbol{x}^0 = (x_1^0,...,x_n^0)$ is called the starting point and the size of this simplex is determined by the stepsizes $\{c_1,...,c_n\}$. The parameters (a, b, d, g) are commonly set to $(1, 0.5, 0.5, 2)$ (Barton and Ivey, 1996).

Barton and Ivey also studied modified NMSM algorithms. The most successful modification re-evaluates the objective function in the best vertex at each shrink step and reduces the simplex by 10% (δ= 0.9) at each shrink step rather than 50% (δ= 0.5). They found that this algorithm, which we will denote with NMSM2, leads to small improvements in the expected value of the objective function at termination at the cost of more function evaluations (Barton and Ivey, 1996). We tested both the original algorithm (denoted with NMSM1), and the modified algorithm.

2.2 Response Surface Methodology

RSM is a collection of statistical and mathematical techniques useful for optimizing stochastic functions (Myers and Montgomery, 1995). The methodology is based on approximation of the objective function by a low order polynomial on a small subregion of the feasible region *D*. The coefficients of the polynomial are determined by regression analysis applied to a number of observations of the objective function. To this end, the objective function is evaluated in an arrangement of points referred to as an experimental design (Kleijnen, 1997). Based on the fitted polynomial, local best values of the parameters $x_1,...,x_n$ are derived, which represent the center point of the new subregion (Fu, 1994).

In the absence of a consensus standard algorithm we attempt in this paper to construct a standard RSM algorithm for automated optimization based on prevailing principles and ideas found in literature. This algorithm comprises two phases: a first-order phase in which first-order polynomials are fitted iteratively until a plateau is reached, or until too much curvature is found (Cochran and Cox, 1962), and a second-order phase in which the objective function is approximated iteratively by second-order polynomials.

The algorithm starts with constructing the first subregion $[x_1^0 - c_1, x_1^0 + c_1]\times...\times[x_n^0 - c_n, x_n^0 + c_n]$ using the starting values of the parameters $\boldsymbol{x}^0 = (x_1^0,...,x_n^0)$ and the initial stepsizes $\{c_1,...,c_n\}$. The parameters are scaled between –1 and +1 such that the subregion corresponds to $[-1,1]\times...\times[-1,1]$ to avoid numerical problems that may occur when parameters vary in orders of magnitude (Free et al., 1987). In the subregion we fit a first-order polynomial represented by

$$\hat{y} = b_0 + \sum_{i=1,...,n} b_i x_i$$

where $\hat{x} \ = (x_1,...,x_n)$ are the scaled parameters. To this end the objective function is evaluated in the 2^n points of a 2-level factorial design, given by the factorial points $(x_1^0 \pm c_1, x_2^0 \pm c_2,..., x_n^0 \pm c_n)$ (Myers and Montgomery, 1995). If the design is not within the feasible region *D*, then it is moved into this region (Smith, 1979). Since we will investigate the objective function for presence of curvature, the objective function is evaluated four times in the center point $(x_1^0,...,x_n^0)$ for testing for lack of fit (Myers and Montgomery, 1995). If there is no systematic curvature present at a 5% significance level, then we test for the presence of a plateau, i.e. we test the hypothesis $H_0 : b_1 = ... = b_n = 0$ against the alternative hypothesis $H_1 : \exists i : b_i \neq 0$. If the null hypothesis is rejected at a 5% significance level then we accept the first-order polynomial and we conclude that a steepest descent direction exists.

In this case, a line search is performed in the steepest descent direction given by $(-b_1,...,-b_n)$ (Myers and Montgomery, 1995). A number of equidistant points in the steepest descent direction will be evaluated, starting at scaled distance 1 from the center point. As soon as a boundary of the feasible region *D* is crossed, the line search is continued along the projection of the search direction on this boundary (Smith, 1979). The line search is ended when an observed value of the simulation response function is higher than the preceding observation. The last point for which the simulation response function was decreasing will be the center point of the next subregion, where again a first-order polynomial is fitted.

If the first-order polynomial is not accepted, then a second-order polynomial is fitted in the current subregion. We use a central composite design (CCD) for determining the coefficients of the second-order polynomial (Kleijnen, 1975), consisting of the center point $(x_1^0,...,x_n^0)$ which is evaluated four times, 2^n scaled factorial points $(x_1^0 \pm c_1, x_2^0 \pm c_2,..., x_n^0 \pm c_n)$ and $2n$ scaled axial points $(x_1^0 \pm a c_1, 0,...,0),...,(0,...,0, x_n^0 \pm a c_n)$ where $a = 2^{n/4}$ (Box and Draper, 1987). A CCD is widely used for fitting second-order polynomials (Myers et al., 1989). The fitted polynomial is represented by

$$\hat{y} = b_0 + \sum_{i=1,...,n} b_i x_i + \sum_{\substack{i,j=1,...,n \\ i \leq j}} b_{ij} x_i x_j = b_0 + \boldsymbol{?'b} + \boldsymbol{?'B?}$$

where $\boldsymbol{?} = (x_1,...,x_n)$ are the scaled parameters. We do not test this polynomial for lack of fit.

The stationary point of the quadratic surface is determined by

$$\boldsymbol{s} = -\frac{1}{2} B^{-1} b$$

Let ***E*** be the $n \times n$ matrix of normalized eigenvectors of ***B*** and let $n_1,...,n_n$ be the eigenvalues of ***B***. If all eigenvalues are positive, then the quadratic surface has a minimum at the stationary point. If this point lies within the current subregion, then it is taken as the center point of the new subregion, whereas the stepsizes $\{c_1,...,c_n\}$ that are used for construction of the subregion are decreased by 50%. In the new subregion again a second-order polynomial will be fitted.

If all eigenvalues are positive but if the stationary point lies outside the current subregion, the stationary point is not regarded as the center of the next subregion. The same applies when the eigenvalues are mixed in sign, i.e. the stationary point is a saddle point or when all eigenvalues are negative, i.e. the stationary point is a maximum.

In this case, *ridge analysis* is performed, which means that we search for a stationary point $\boldsymbol{?}_{\boldsymbol{R}}$ on a given radius R such that the quadratic surface has a minimum at this stationary point (Myers and Montgomery, 1995). Using Lagrange analysis with multiplier m, this stationary point is given by

$$(\boldsymbol{B} - m\boldsymbol{I})\boldsymbol{?}_{\boldsymbol{R}} = -\boldsymbol{b}/2$$

and it should hold that $m < \min_i n_i$ and $\sqrt{\boldsymbol{?}'_{\boldsymbol{R}}\boldsymbol{?}_{\boldsymbol{R}}} = R$. We can write

$$R^2 = \boldsymbol{?}'_{\boldsymbol{R}}\boldsymbol{?}_{\boldsymbol{R}} = \sum_{i=1}^{n}\left(\frac{\boldsymbol{e}'_i\boldsymbol{b}}{2(n_i - m)}\right)^2$$

where $\boldsymbol{e}_i$ is the eigenvector corresponding to the i^{th} eigenvalue n_i. We consider the radius of the circumscribed sphere of the subregion, i.e. $R = \sqrt{2}$, which means that we have to find $m < \min_i n_i$ such that

$$\sum_{i=1}^{n}\left(\frac{\boldsymbol{e}'_i\boldsymbol{b}}{2(n_i - m)}\right)^2 = 2$$

Standard numerical methods for finding the root of an equation are used to determine m. The stationary point that results from the ridge analysis will be used as the center point of the next subregion, in which again a second-order polynomial will be fitted.

The algorithm described above is referred to as RSM1. To study some of the choices we made in constructing this algorithm we also investigate the following algorithms, which are equal to RSM1 except for a single modification:

- RSM2: if the stationary point is a minimum inside the subregion, then the stepsizes $\{c_1,...,c_n\}$ are decreased by 10% instead of by 50%.
- RSM3: the significance level used for statistically testing of the first-order polynomial equals 2.5% instead of 5%.
- RSM4: the stepsizes $\{c_1,...,c_n\}$ are decreased only if for two consecutive second-order polynomials the stationary point is a minimum inside the subregion, instead of decreasing the stepsizes as soon as a minimum is found inside the subregion.

3 Test problems

We test the six optimization algorithms on a set of 18 deterministic test functions for unconstrained optimization, that were made stochastic by adding random noise. Barton and Ivey (1996) also used these functions in comparing different versions of the NMSM algorithm. As it is not clear whether these functions represent real microsimulation models we also consider a microsimulation version of an existing cancer screening model. This model has three parameters that need to be estimated from an observed data set by constrained minimization of a goodness-of-fit test statistic. For this particular model the optimal parameters can also be determined analytically.

3.1 One stage-one test breast cancer model

The microsimulation model is a simulation implementation of the breast cancer screening model developed by Day and Walter (1984). In this model only one disease stage, the detectable preclinical phase (DPCP), is modeled. The DPCP has incidence rate J and we assume that the duration of the DPCP is exponentially distributed with parameter l. At the end of the DPCP a cancer is clinically detected, whereas during the DPCP a cancer can be detected by breast cancer screening.

A screening programme consisting of four annual screening rounds is simulated. The sensitivity of the screening test is denoted with j . In each microsimulation run 50,000 individual life histories, including the disease processes and the impact of screening, are simulated. The microsimulation model simulates detection rates at each of the screening rounds and incidence rates of clinical disease in the period following a negative screening test, for each of the screening rounds and for different intervals since the screening test.

The model will be applied to data from the first randomized trial for breast cancer screening, viz. the HIP study (Day and Walter, 1984; Shapiro et al., 1974; van Oortmarssen et al., 1990). In the HIP study approximately 62,000 women, who were aged between 40 and 64 at entry, were randomly allocated to either a study group or a control group. Only the study group was offered annual breast cancer screening for four years. About 65 percent of the study group (20,166 women) agreed to take part and were screened at least once (these women all attended the first screening). We will use follow-up data until 5 years after the last screening. The results from the HIP screening trial that will be used are described by Day and Walter (1984), and consists of 4 detection rates and 14 incidence rates of interval cancers occurring after a previous negative test result.

The parameters $J,\ l$ and j will be estimated from the observed data set through minimization of a chi-square goodness-of-fit test statistic. The simulation response function is given by

$$F(J,\ l, j) = \sum_{i=1}^{18} (O_i(J,\ l, j) - E_i(J,\ l, j))^2 / E_i(J,\ l, j)$$

where O_i is the observed number of cancers during screening round or interval *i* and E_i is the number of simulated cancers during screening round or interval *i*, $i = 1,\ldots,18$. The true optimal parameters of the model for the HIP data were derived using the objective function

$$f(J,\ l, j) = \sum_{i=1}^{18} (O_i(J,\ l, j) - A_i(J,\ l, j))^2 / A_i(J,\ l, j)$$

where A_i is the number of cancers during screening round or interval *i*, $i = 1,\ldots,18$, as predicted by the analytical implementation of the breast cancer screening model (Day and Walter, 1984). We determined the optimal parameters $(J^*,\ l^*, j^*)$ of the model applied to the HIP data set by extensive enumeration (using the stepsizes 10^{-5} for J, 10^{-4} for l and 10^{-5} for j) of $f(J,\ l, j)$:

$$f(J^*,\ l^*, j^*) = f(0.0021, 0.620, 0.8760) \approx 13.4696 .$$

Starting values, constraints and initial stepsizes for the parameters $J,\ l$ and j that were used for the minimization of the goodness-of-fit test statistic are given in Table 1.

3.2 The test functions

In addition to the HIP screening model we test randomized versions of 18 deterministic unconstrained nonlinear minimization problems that were suggested by Moré et al (1981). In their paper, the starting points that are used in this study, and most of the optimal solutions of the test problems are given. The test functions can also be found in the NETLIB collection (http://www.netlib.org/uncon/). In case the dimensions of the test functions can be varied, we use the same dimensions as Barton and Ivey (1996). In Table 2 the dimensions, the optimal solutions as well as the starting points used in the optimization of the 18 test functions are given. For each test function we use initial stepsizes $\{c_1,\ldots,c_n\} = (1,\ldots,1)$.

Following Barton and Ivey we make the 18 test functions stochastic by adding a standard normal distributed random variable truncated to ±3. Independent random number streams are used for each optimization run.

4 Experiments and statistical analysis

For both the microsimulation model and each of the test functions, we performed twenty optimization runs with each optimization algorithm. For the test functions, optimization runs were terminated after 5000 function evaluations for both the NMSM algorithms and the RSM algorithms (Barton, 1987). The optimization of the microsimulation model was terminated after 2000 function evaluations, since the number of parameters is low and the evaluation of the objective function is rather time-consuming compared to the evaluation of the test functions. In all cases the runs were long enough for the optimization algorithms to stabilize on some value.

For comparing the six algorithms, the observed best parameter values $\boldsymbol{x}_{opt}$ of an optimization run as well as the corresponding value of the simulation response function need to be specified. For the NMSM algorithms $\boldsymbol{x}_{opt}$ is given by the best vertex of the last complete simplex, and the corresponding value of the simulation response function is given by $F(\boldsymbol{x}_{opt})$ as obtained during the optimization run. For the RSM algorithms, $\boldsymbol{x}_{opt}$ is given by the center point of the last subregion for which the objective function has been evaluated four times in the center point. The corresponding value of the simulation response function $F(\boldsymbol{x}_{opt})$ is given by the average value of these four evaluations.

To evaluate the accuracy and reliability of an optimization run, we define the *error* of an optimization run as the difference between the expected simulation response function value $f(\boldsymbol{x}^*)$ in the true optimal point $\boldsymbol{x}^*$ and the expected simulation response function value $f(\boldsymbol{x}_{opt})$ in the observed best point of the run:

$$e = f(\boldsymbol{x}_{opt}) - f(\boldsymbol{x}^*).$$

For each algorithm $i,\ i \in \{NMSM1, NMSM2, RSM1, RSM2, RSM3, RSM4\}$, the errors $e_{i,j},\ j = 1,\ldots,20$ resulting from the twenty optimization runs are mutually independent. We assume that for each algorithm the twenty errors come from the same continuous distribution. However, as can be seen in the next section, the distributions for the six algorithms can be different.

For both the microsimulation model and for each of the test functions, we compare the reliability of the six algorithms by looking at the variance of the errors.

The accuracy of the six algorithms is compared by using nonparametric tests to determine if there is any stochastic difference between the algorithms (Hollander and Wolfe, 1973),

using a 5% significance level. For comparison of the NMSM algorithms with the RSM algorithms we pooled the errors from the two NMSM algorithms as well as the errors from the four RSM algorithms, which results in two sets of errors:
$e_{i,j}$, $\mathrm{i = NMSM1, NMSM2,\ j = 1,...,20}$ and $e_{i,j}$, $\mathrm{i = RSM1,..., RSM4,\ j = 1,...,20}$.
Letting $\Delta = [P(e_{NMSM} < e_{RSM}) - (1/2)]$ we test the hypothesis $H_0 : \Delta = 0$, i.e. there is no stochastic difference between the two sets of algorithms, by using the Wilcoxon rank sum test. We also used this test in a similar way for the mutual comparison of the two NMSM algorithms.

For comparison of the four RSM algorithms we use the Kruskal-Wallis test. Here, the null hypothesis that the errors of the four algorithms are stochastically equal is tested against the alternative hypothesis that at least one of the algorithms performs stochastically different. In case of stochastic difference between the RSM algorithms, algorithms RSM*2*, RSM*3* and RSM*4* are compared to the basic algorithm RSM*1* by using distribution-free multiple comparisons based on the Kruskal-Wallis test.

We also consider the efficiency of the optimization algorithms by looking at the number of function evaluations needed for the best point of an iteration to come within a certain distance of the observed optimum. We define the following measure for each iteration $i, i = 1,..., M$ (Barton, 1984):

$$G_i = \frac{F(\boldsymbol{x}_1) - F(\boldsymbol{x}_i)}{F(\boldsymbol{x}_1) - F(\boldsymbol{x}_{opt})}$$

Here M is the total number of iterations performed in the optimization run. For the NMSM algorithms $F(\boldsymbol{x}_i)$ is given by the simulation response function value in the best vertex of the simplex in iteration *i*. For the RSM algorithms, $F(\boldsymbol{x}_i)$ is given by the average of the four simulation response function values in the center point of the subregion used in iteration *i*.

Since $F(\boldsymbol{x}_1) - F(\boldsymbol{x}_{opt})$ is the gap between the observed starting value of the simulation response function and the simulation response function value in the observed best point of the run, G_i describes the reduction of this gap that is achieved after *i* iterations of this optimization run. It should be noted that due to noise it is possible that $G_i \notin [0,1]$ for some iteration *i*. Moreover, we have $G_1 = 0$ and $G_M = 1$. For each optimization run we consider the first iteration for which G_i exceeds a predefined percentage $q \in [0,1]$. To compare the efficiency of the six algorithms, we define S_q as the cumulative number of evaluations performed up to and including this iteration. We consider the values $q = 0.95$ and $q = 0.99$, which gives for each algorithm twenty values for $S_{0.95,i}, i = 1,...,20$ and twenty values for $S_{0.99,i}, i = 1,...,20$.

5 Evaluation of the results

5.1 Results for the microsimulation model

In Figure 1 the errors resulting from 20 optimization runs for the six algorithms are shown. The variance of the errors across the algorithms NMSM1 and NMSM2 is much higher than the variance across the RSM algorithms, which make the latter more reliable. In many cases the algorithms NMSM1 and NMSM2 performed less accurately than the RSM algorithms. The averages and standard deviations of the errors resulting from the optimization of the microsimulation model are given in Table 3. On average the algorithms RSM2 and RSM4 have the smallest error, and all the RSM algorithms outperform the two NMSM algorithms. Indeed, from the Wilcoxon test we conclude that the RSM algorithms perform stochastically better and thus are more accurate than the NMSM algorithms. Moreover, we find that the modified NMSM algorithm performs stochastically better than the original NMSM algorithm, which is consistent with the findings of Barton and Ivey. The Kruskal-Wallis test indicates that there is stochastic difference between the RSM algorithms. We find that algorithms RSM*2* and RSM*4* perform stochastically better than the basic algorithm RSM*1*. Compared to the basic algorithm RSM*1*, these algorithms differ in the way the stepsizes are decreased during an optimization run.

The first-order polynomial that was fitted in the first subregion of an optimization run was not accepted in all the optimization runs done with all four RSM algorithms. This was the case for both the 2.5% significance level (RSM*3*) and the 5% significance level (RSM*1*, RSM*2*, RSM*4*), which explains why there is no stochastic difference between the basic algorithm RSM*1* and algorithm RSM*3*.

Figures 2a - 2f show the observed best function value in an iteration as function of the cumulative number of evaluations performed up to and including this iteration for each of the 20 optimization runs and for each of the six algorithms. The averages and standard deviations of the efficiency measures $S_{0.95}$ and $S_{0.99}$ are given in Table 4. On average algorithm NMSM2 converges much faster to the observed optimum than algorithm NMSM1. For algorithm NMSM1 the variance of $S_{0.95}$ and $S_{0.99}$ is quite large, which is explained by the many cases in which the observed best function value gets stuck during the optimization run but improves in one of the last iterations of the run, as can be seen in Figure 2a. Furthermore, on average the NMSM algorithms are faster than the RSM algorithms, and RSM*1* is the fastest RSM algorithm.

We conclude that for the microsimulation model the modification in the original NMSM algorithm both leads to higher accuracy, higher reliability and more efficiency. The RSM algorithms perform more accurately and reliable than the NMSM algorithms. However, the RSM algorithms are considerably less efficient. The modified RSM algorithms RSM*2* and RSM*4* perform more accurately than RSM*1*, at the cost of lower efficiency.

5.2 Results for the test functions

In Table 5 the averages and standard deviations of the errors resulting from the optimization of the test functions are given, and Table 6 shows the statistical results for the six algorithms and for the 18 test functions. With respect to the reliability of the algorithms we find that the results are mixed: for some test functions the errors resulting from the NMSM algorithms are much more variable than the errors resulting from the RSM algorithms, but the reverse can also be found. We find that for most test functions the modified NMSM algorithm is more reliable than the original NMSM algorithm. We find that for 11 test functions the NMSM algorithms performed clearly more accurately than the RSM algorithms, whereas for 6 test functions the RSM algorithms performed more accurately than the NMSM algorithms. For some test functions the RSM algorithms performed very bad, as can be seen in Table 5. Like Barton and Ivey we found that the modified NMSM algorithm has a better performance than the original algorithm NMSM*1*, although this difference was not statistically significant for some test functions. For eight test functions we found a stochastic difference between the RSM functions. We find that in most cases algorithm RSM*1* performs stochastically better than the algorithms RSM*2* and RSM*4*, whereas no difference is found between RSM*1* and RSM*3*. Again we find for all RSM algorithms that the first-order polynomial that was fitted in the first subregion in most cases was not accepted.

The averages and standard deviations of the efficiency measures $S_{0.95}$ and $S_{0.99}$ are given in Tables 7 and 8. Again it can be seen that on average algorithm NMSM2 converges faster to its observed optimum than algorithm NMSM1, and that both NMSM algorithms converge faster than the RSM algorithms. Moreover, for some test functions and algorithms the variances of $S_{0.95}$ and $S_{0.99}$ are quite large.

We conclude that also for the test functions the modification in the original NMSM algorithm both leads to higher accuracy, higher reliability and more efficiency. We did not find an algorithm that outperformed the other algorithms for all the test functions with regard to accuracy or reliability. However, we did find that the RSM algorithms are considerably less efficient than the NMSM algorithms.

6 Conclusions and Further Research

The growing demand for complex microsimulation models requires continuous efforts to devise and test robust and efficient optimization methods. Barton and Ivey (1996) investigated NMSM for its use in simulation optimization. They reported that a modified NMSM algorithm performed better than the original method. However, the performance of this method in comparison to alternative methods such as RSM still had to be established. This study is the first attempt to compare the performance of NMSM and RSM using a standard set of difficult test functions, and a representative (albeit simple) microsimulation model.

Both with respect to a microsimulation version of a cancer screening model and a set of test functions we found that the modified NMSM algorithm performed more accurately and reliable than its original. In contrast to the findings of Barton and Ivey (1996), we found that this modified algorithm performed more efficiently than the original algorithm. However, this is probably caused by the use of different criterions for ending the optimization runs.

With respect to the microsimulation model we found that the tested RSM algorithms performed more accurately than the two NMSM algorithms, although the NMSM algorithms did show rather accurate results in some optimization runs. The reliability of the NMSM algorithms is rather low compared to the RSM algorithms. Using multistart when optimizing with a NMSM algorithm in combination with increasing the simulation size to decrease the noise coming from the microsimulation model, could probably prevent highly inaccurate optimization results and increase the reliability.

No consistent results were found for the test functions. None of the algorithms performed satisfactorily on all functions with respect to accuracy, although the NMSM algorithms often outperformed the RSM algorithms. The test functions are difficult and show rather erratic behavior. For some of the test functions, the observed best function value gets stuck far away from the real optimum when using the RSM algorithms. The NMSM algorithms seem to be better suited for this erratic behavior.

For both the microsimulation model and the test functions, the RSM algorithms were clearly less efficient than the NMSM algorithms. The slower convergence of the RSM algorithms is first of all caused by the large size of its designs. Since each point of the design induces a function evaluation, which in case of microsimulation can be very time consuming, designs with fewer points, such as fractional factorial designs, could be more efficient. Moreover, the first-order polynomial that was fitted during the first iteration of any RSM algorithm was almost always rejected. This means that during an optimization run only second-order polynomials are considered, without the possibility of returning to the first-order phase. However, during this first-order phase large steps towards the optimum can be taken due to the use of line search, whereas during the second-order phase only steps are being made within a subregion. Therefore, it could be explored whether it is more efficient if in a certain fixed number of subregions at the start of the optimization run only first-order polynomials are fitted before going to the second-order phase, or if returning to the first-order phase would be allowed.

In addition to considering possible improvements of the RSM and NMSM algorithms, the question how these algorithms compare to other algorithms such as Stochastic Approximation and Simultaneous Perturbation Stochastic Approximation remains to be addressed.

References

Barton R.R., 1984. Minimization Algorithms for Functions with Random Noise. American Journal of Mathematical and Management Sciences 4, 109-138.

Barton R.R., 1987. Testing Strategies for Simulation Optimization. Proceedings of the 1987 Winter Simulation Conference, 391-401.

Barton R.R., Ivey J.S., 1996. Nelder-Mead Simplex Modifications for Simulation Optimization. Management Science 42, 954-973.

Box G.E.P., Draper N.R., 1987. Empirical Model-Building and Response Surfaces, John Wiley & Sons, New York.

Cochran W.G., Cox G.M, 1962. Experimental Designs, John Wiley & Sons, New York.

Day N.E., Walter S.D., 1984. Simplified Models of Screening for Chronic Disease: Estimation Procedures from Mass Screening Programmes. Biometrics 40,1-14.

Dennis J.E., Woods D.J., 1987. Optimization on Microcomputers: The Nelder-Mead Simplex Algorithm. In: Wouk A., New Computing Environments: Microcomputers in Large Scale Computing, SIAM, Philadelphia, pp.116-122.

Free J.W., Parkinson A.R., Bryce G.R., Balling R.J., 1987. Approximation of Computationally Expensive and Noisy Functions for Constrained Nonlinear Optimizations. Transactions of the American Society of Mechanical Engineers 109, 528-532.

Fu M.C., 1994. Optimization via Simulation: A Review. Annals of Operations Research 53, 199-247.

Hollander M., Wolfe D.A., 1973. Nonparametric Statistical methods. John Wiley & Sons, New York.

Jacobson S.H., Schruben L.W., 1989. Techniques for Simulation Response Optimization. Operation Research Letters 8, 1-9.

Kleijnen, J.P.C. 1974. Statistical Techniques in Simulation: Part I, Marcel Dekker, New York.

Kleijnen, J.P.C. 1975. Statistical Techniques in Simulation: Part II, Marcel Dekker, New York.

Kleijnen, J.P.C., 1997. Experimental Design for Sensitivity Analysis, Optimization, and Validation of Simulation models. In: Banks, J., Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice, John Wiley & Sons, New York, pp.173-223.

Moré J.J., Garbow B.S., Hillstrom K.E., 1981. Testing Unconstrained Optimization Software. ACM Transactions on Mathematical Software 7 (1), 17-41.

Myers R.H., Khuri A.I., Carter W.H., 1989. Response Surface Methodology: 1966-1988. Technometrics 31, 137-157.

Myers R.H., Montgomery D.C., 1995. Response Surface Methodology: Process and Product Optimization Using Designed Experiments, John Wiley & Sons, New York.

Nelson L.S., 1985. Nelder-Mead Simplex Method. In: Kotz S. and Johnson N.L. (Eds.), Encyclopedia of Statistical Sciences, John Wiley & Sons, New York, pp. 178-181.

Oortmarssen G.J. van, Habbema J.D.F., Lubbe J.Th.N., Maas P.J. van der, 1990. A Model-based Analysis of the HIP Project for Breast Cancer Screening. International Journal of Cancer 46, 207-213.

Pflug G.Ch., 1996. Optimization of Stochastic Models: The Interface Between Simulation and Optimization, Kluwer Academic Publishers, Boston.

Shapiro S., Goldberg J.D., Hutchison G.B., 1974. Lead Time in Breast Cancer Detection and Implications for Periodicity of Screening. American Journal of Epidemiology 100, 357-366.

Smith D.E., 1979. Automatic Optimum-Seeking Program for Digital Simulation. Simulation 27, 27-32.

Spall J.C., 1992. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. IEEE Transactions on Automatic Control 37 (3), 332-341.

Table 1 Starting values, stepsizes and constraints for the microsimulation model parameters.

Parameter	Start	Lower limit	Upper limit	Stepsize
Incidence rate J	0.002	0.00002	0.02	0.0001
Duration parameter l (mean duration = $1/ l$)	0.6	0	1	0.01
Sensitivity j	0.6	0	1	0.01

Table 2 The 18 test functions used for testing the NMSM and RSM algorithms.

Test function		Dimension	Starting point	Optimal value
1.	Helical valley function	3	(-1,0,0)	0
2.	Biggs Exp6 function	6	(1,2,1,1,1,1)	0
3.	Gaussian function	3	(0.4,1,0)	1.12793…e-8
4.	Powell badly scaled function	2	(0,1)	0
5.	Box 3-dimensional function	3	(0,10,20)	0
6.	Variably dimensioned function	4	$(3/4, 1/2, 1/4, 0)$	0
7.	Watson function	9	(0,…,0)	1.39976…e-6
8.	Penalty function I	8	(1,2,3,4,5,6,7,8)	5.42152…e-5*
9.	Penalty function II	8	$(1/2, ..., 1/2)$	1.23335…e-4*
10.	Brown badly scaled function	2	(1,1)	0
11.	Brown and Dennis function	4	(25,5,-5,-1)	85822.2…
12.	Gulf research and development function	3	(5,2.5,0.15)	0
13.	Trigonometric function	8	$(1/8, ..., 1/8)$	0
14.	Extended Rosenbrock function	4	(-1.2,1,-1.2,1)	0
15.	Extended Powell function	8	(3,-1,0,1,3,-1,0,1)	0
16.	Beale function	2	(1,1)	0
17.	Wood function	4	(-3,-1,-3,-1)	0
18.	Chebyquad function	8	$(1/9, ..., 8/9)$	3.51687…e-3

*) This optimum was found by using the NEOS server (http://www-unix.mcs.anl.gov/neos/Server). We used the NMTR routine (a trust region version of Newton's method).

Table 3 Averages and standard deviations of the errors resulting from the optimization of the microsimulation model. An error is defined as the difference between the expected simulation response function value in the true optimal point and the expected simulation response function value in the observed best point of an optimization run.

Optimization Algorithm	NMSM1	NMSM2	RSM1	RSM2	RSM3	RSM4
Average error (st.dev. error)	6.54 (4.80)	3.27 (4.05)	0.57 (0.23)	0.26 (0.16)	0.72 (0.29)	0.20 (0.14)

Table 4 Averages and standard deviations of the efficiency measures $S_{0.95}$ and $S_{0.99}$ (as defined in Section 4) of the optimization of the microsimulation model.

Optimization Algorithm	Average of $S_{0.95}$ (standard deviation of $S_{0.95}$)	Average of $S_{0.99}$ (standard deviation of $S_{0.99}$)
NMSM1	198 (262)	612 (574)
NMSM2	49 (20)	67 (32)
RSM1	218 (19)	329 (122)
RSM2	253 (38)	531 (218)
RSM3	230 (34)	328 (211)
RSM4	280 (84)	491 (251)

Table 5 Averages and standard deviations of the errors resulting from the optimization of the test functions. An error is defined as the difference between the expected simulation response function value in the true optimal point and the expected simulation response function value in the observed best point of an optimization run.

	Optimization Algorithm					
Test Function	NMSM1	NMSM2	RSM1	RSM2	RSM3	RSM4
1.	0.58 (0.29)	0.48 (0.34)	0.19 (0.10)	21.48 (10.68)	0.23 (0.14)	13.73 (21.75)
2.	0.52 (0.17)	0.42 (0.09)	1.00 (0.52)	1.19 (0.20)	1.00 (0.47)	1.27 (0.22)
3.	0.21 (0.20)	0.18 (0.16)	0.33 (0.21)	0.47 (0.15)	0.31 (0.21)	0.53 (0.12)
4.	0.52 (0.68)	0.27 (0.19)	1.13 (0.23)	1.12 (0.17)	1.10 (0.18)	1.10 (0.11)
5.	0.35 (0.36)	0.22 (0.15)	0.15 (0.11)	0.17 (0.10)	0.20 (0.24)	0.10 (0.11)
6.	0.28 (0.32)	0.19 (0.16)	6.88 (0.00)	6.88 (0.00)	6.88 (0.00)	6.88 (0.00)
7.	0.48 (0.22)	0.23 (0.14)	143.77 (1.94)	143.42 (1.78)	144.20 (2.01)	144.22 (2.04)
8.	0.58 (0.46)	0.35 (0.32)	0.01 (0.01)	0.06 (0.00)	0.02 (0.02)	0.01 (0.01)
9.	0.16 (0.15)	0.12 (0.10)	0.18 (0.10)	0.78 (0.09)	0.18 (0.10)	0.20 (0.10)
10.	0.96 (0.58)	0.75 (0.56)	9.99E+11 (0.00)	9.99E+11 (0.00)	9.99E+11 (0.00)	9.99E+11 (0.00)
11.	0.36 (0.24)	0.25 (0.17)	0.29 (0.20)	0.21 (0.13)	0.21 (0.11)	0.21 (0.16)
12.	6.92 (0.43)	6.73 (0.16)	6.65 (0.14)	6.69 (0.25)	6.66 (0.12)	6.37 (0.24)
13.	0.2199 (0.18)	0.1520 (0.07)	0.7285 (0.11)	0.73 (0.12)	0.70 (0.10)	0.69 (0.15)
14.	9.05 (0.30)	8.85 (0.26)	120.99 (24.74)	131.99 (33.19)	113.65 (30.84)	126.57 (29.72)
15.	9.68 (1.40)	8.79 (1.64)	0.49 (0.17)	0.75 (0.09)	0.40 (0.11)	0.30 (0.09)
16.	0.75 (0.60)	0.41 (0.41)	0.42 (0.33)	0.20 (0.22)	0.28 (0.27)	0.12 (0.12)
17.	8.36 (0.35)	8.12 (0.19)	95.81 (22.43)	89.01 (16.27)	87.06 (15.03)	84.24 (14.47)
18.	0.57 (0.23)	0.30 (0.12)	0.21 (0.04)	3.55 (0.00)	0.22 (0.05)	0.27 (0.06)

Table 6 Results from the comparison of the six algorithms using nonparametric statistical methods

- NMSM/RSM: Comparison NMSM and RSM algorithms: preferred method
- NMSM: Comparison algorithms NMSM1 and NMSM2: preferred method
- RSM: Comparison RSM algorithms: do the four algorithms perform stochastically different?
- RSM2/RSM1: Comparison RSM1 and RSM2: preferred method
- RSM3/RSM1: Comparison RSM1 and RSM3: preferred method
- RSM4/RSM1: Comparison RSM1 and RSM4: preferred method

N.B. '---' means that there was no preferred method, 'n/a' means that the nonparametric statistical method was not applicable.

Test function	NMSM / RSM	NMSM	RSM	RSM2 / RSM1	RSM3 / RSM1	RSM4 / RSM1
1.	---	---	YES	RSM1	---	RSM1
2.	NM	NMSM2	---	n/a	n/a	n/a
3.	NM	---	YES	---	---	---
4.	NM	---	---	n/a	n/a	n/a
5.	RSM	---	---	n/a	n/a	n/a
6.	NM	---	---	n/a	n/a	n/a
7.	NM	NMSM2	---	n/a	n/a	n/a
8.	RSM	NMSM2	YES	RSM1	---	---
9.	NM	---	YES	RSM1	---	---
10.	NM	---	---	n/a	n/a	n/a
11.	RSM	---	---	n/a	n/a	n/a
12.	RSM	---	YES	---	---	RSM4
13.	NM	---	---	n/a	n/a	n/a
14.	NM	NMSM2	---	n/a	n/a	n/a
15.	RSM	NMSM2	YES	RSM1	---	RSM4
16.	RSM	NMSM2	YES	RSM2	---	RSM4
17.	NM	NMSM2	---	n/a	n/a	n/a
18.	NM	NMSM2	YES	RSM1	---	RSM1

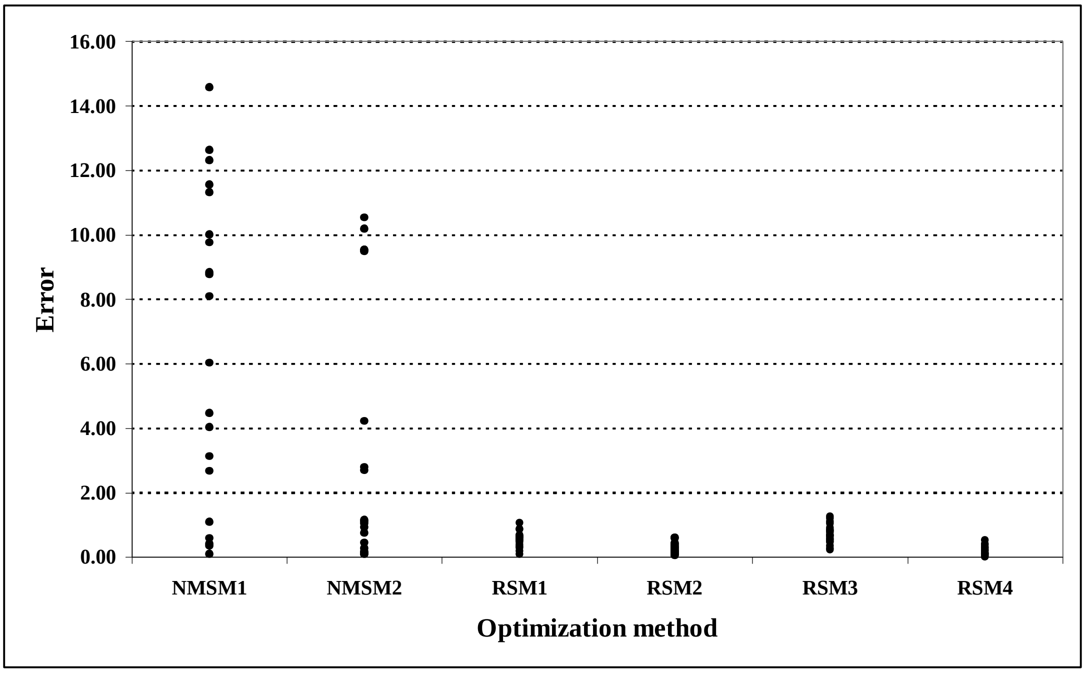
Table 7 Averages and standard deviations of the efficiency measure $S_{0.95}$ (as defined in Section 4) resulting from the optimization of the test functions.

	Optimization Algorithm					
Test function	NMSM1	NMSM2	RSM1	RSM2	RSM3	RSM4
1.	9 (0)	9 (0)	36 (0)	36 (0)	36 (0)	36 (0)
2.	715 (653)	299 (1107)	396 (504)	866 (1201)	1352 (1634)	754 (1092)
3.	411 (351)	144 (308)	120 (94)	276 (473)	263 (480)	97 (82)
4.	702 (893)	310 (1105)	212 (210)	211 (469)	148 (151)	188 (262)
5.	17 (0)	17 (0)	216 (0)	216 (0)	216 (0)	216 (0)
6.	7 (0)	7 (0)	56 (0)	56 (0)	56 (0)	56 (0)
7.	250 (215)	105 (76)	2670 (755)	2590 (818)	2884 (834)	2884 (873)
8.	111 (0)	111 (0)	1932 (0)	1932 (0)	1932 (0)	1932 (0)
9.	69 (4)	64 (5)	842 (62)	552 (0)	869 (101)	552 (0)
10.	157 (0)	157 (0)	4764 (0)	4764 (0)	4764 (0)	4764 (0)
11.	65 (0)	65 (0)	448 (0)	448 (0)	448 (0)	448 (0)
12.	218 (162)	22 (16)	93 (29)	316 (213)	91 (39)	238 (266)
13.	277 (210)	379 (1119)	912 (1055)	802 (992)	593 (101)	1259 (1400)
14.	43 (10)	32 (5)	84 (0)	84 (0)	83 (6)	84 (0)
15.	44 (3)	45 (3)	1587 (123)	2484 (0)	1559 (135)	1656 (0)
16.	102 (74)	19 (18)	117 (37)	108 (33)	203 (127)	167 (70)
17.	13 (0)	13 (0)	84 (0)	84 (0)	84 (0)	84 (0)
18.	384 (297)	128 (232)	1466 (1346)	1727 (802)	1549 (1350)	1742 (1833)

Table 8 Averages and standard deviations of the efficiency measure $S_{0.99}$ (as defined in Section 4) resulting from the optimization of the test functions.

	Optimization Algorithm					
Test function	NMSM1	NMSM2	RSM1	RSM2	RSM3	RSM4
1.	21 (26)	20 (1)	108 (0)	50 (24)	108 (0)	82 (20)
2.	821 (666)	302 (1107)	424 (546)	922 (1202)	1412 (1665)	942 (1396)
3.	617 (702)	154 (315)	130 (115)	302 (555)	419 (893)	97 (82)
4.	844 (859)	319 (1104)	232 (279)	211 (469)	153 (156)	210 (267)
5.	29 (3)	27 (3)	252 (0)	252 (0)	252 (0)	252 (0)
6.	34 (16)	13 (7)	84 (0)	84 (0)	84 (0)	84 (0)
7.	670 (842)	158 (207)	4073 (748)	4000 (683)	4199 (749)	4226 (741)
8.	128 (0)	128 (0)	2208 (0)	2208 (0)	2208 (0)	2208 (0)
9.	150 (45)	98 (34)	1187 (130)	662 (208)	1242 (385)	690 (443)
10.	161 (0)	161 (0)	4956 (0)	4956 (0)	4956 (0)	4956 (0)
11.	76 (0)	76 (0)	644 (0)	644 (0)	644 (0)	644 (0)
12.	677 (647)	28 (27)	95 (32)	356 (215)	93 (38)	469 (751)
13.	432 (401)	388 (1121)	912 (1055)	802 (992)	593 (101)	1259 (1400)
14.	217 (238)	58 (30)	84 (0)	84 (0)	83 (6)	84 (0)
15.	85 (11)	94 (16)	3119 (221)	4181 (101)	3119 (239)	3091 (212)
16.	393 (332)	30 (29)	129 (48)	126 (41)	346 (476)	205 (128)
17.	22 (0)	22 (0)	112 (0)	112 (0)	112 (0)	112 (0)
18.	467 (366)	177 (330)	1466 (1346)	1809 (825)	1744 (1549)	1742 (1833)

Figure 1 Errors resulting from 20 optimization runs for the microsimulation model, for each of the six optimization algorithms. An error is defined as the difference between the expected simulation response function value in the true optimal point and the expected simulation response function value in the observed best point of an optimization run.



Figures 2a - 2f Observed best function value in an iteration (as defined in Section 4) as function of the cumulative number of evaluations performed up to and including this iteration, resulting from 20 optimization runs of the microsimulation model with each of the six algorithms.

These figures are available on request from the first author.