TI 2005-042/4
Tinbergen Institute Discussion Paper

Automated Response Surface Methodology for Stochastic Optimization Models with Unknown Variance

Robin P. Nicolai
Rommert Dekker

Faculty of Economics, Erasmus Universiteit Rotterdam, and Tinbergen Institute.

Tinbergen Institute
The Tinbergen Institute is the institute for economic research of the Erasmus Universiteit Rotterdam, Universiteit van Amsterdam, and Vrije Universiteit Amsterdam.

Tinbergen Institute Amsterdam
Roetersstraat 31
1018 WB Amsterdam
The Netherlands
Tel.: +31(0)20 551 3500
Fax: +31(0)20 551 3555

Tinbergen Institute Rotterdam
Burg. Oudlaan 50
3062 PA Rotterdam
The Netherlands
Tel.: +31(0)10 408 8900
Fax: +31(0)10 408 9031

Please send questions and/or remarks of non-scientific nature to driessen@tinbergen.nl.
Most TI discussion papers can be downloaded at http://www.tinbergen.nl.

Automated Response Surface Methodology for Stochastic Optimization Models with Unknown Variance

Robin Nicolai[a,b,*], Rommert Dekker[a,b]

[a]Department of Econometrics and Operations Research, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
[b]Tinbergen Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
[*]Corresponding author. Tel. +31 10 4082524; email rnicolai@few.eur.nl.

Abstract
Response Surface Methodology (RSM) is a tool that was introduced in the early 50´s by Box and Wilson (1951). It is a collection of mathematical and statistical techniques useful for the approximation and optimization of stochastic models. Applications of RSM can be found in e.g. chemical, engineering and clinical sciences. In this paper we are interested in finding the best settings for an automated RSM procedure when there is very little information about the stochastic objective function. We will present a framework of the RSM procedures for finding optimal solutions in the presence of noise. We emphasize the use of both stopping rules and restart procedures. Good stopping rules recognize when no further improvement is being made. Restarts are used to escape from non-optimal regions of the domain. We compare different versions of the RSM algorithms on a number of test functions, including a simulation model for cancer screening. The results show that considerable improvement is possible over the proposed settings in the existing literature.

Keywords
Response Surface Methodology; Simulation Optimization

JEL Classification
C61

1 Introduction

Response Surface Methodology (RSM) is a tool that was introduced in the early 50´s by Box and Wilson (1951). It is a collection of mathematical and statistical techniques that is useful for the approximation and optimization of stochastic functions. RSM is based on approximations of the objective function by a low order polynomial on a small sub-region of the domain. Using regression analysis based on a number of observations of the stochastic objective function, the best local solution is determined together with a search direction for possible improvement. To this end, the stochastic function is evaluated in a specific arrangement of points referred to as an experimental design. Many applications of the RSM procedure are performed in a manual setting, for example in physical, engineering, biological, clinical and food sciences (Myers et al., 1989).

In a manual setting the user can interfere in the optimization process according to his/her personal intuition and likings. In an automated RSM optimization exercise the settings of the algorithm have to be fixed in a systematic manner. We want to design a RSM algorithm that does not stop to ask for input from the user during an optimization run, instead the algorithm reads the input, performs a systematic search for an optimum and reports the optimum back to the user. The OPTQUEST (Glover et al. 1999) simulation optimization procedure operates in similar way, yet it is primarily oriented at optimization of discrete decision variables and it uses other techniques.

In this paper we are interested in finding the best settings for such an automated RSM algorithm when there is very little information about the objective function. We consider stochastic objective functions with unknown variance that are somewhat time-consuming to evaluate for each solution. When optimizing a simulation model, one estimates the model parameters that optimize specific stochastic output statistics of the simulation model. In this optimization exercise the simulation model is considered to be a black box. The advantage of such a procedure is that the original simulation model can be left intact, while procedures like infinitesimal perturbation require software changes, that are not desirable in case of very complex models.

For an automated RSM to be called successful it should be reliable, precise and fast. The procedure should recognize when no further progress is being made and the differences in the subsequent iterations can only be attributed to the noise in the objective function. Moreover, the procedure should also be able to distinguish optimal solutions from random fluctuations.

In this paper we will present a framework of RSM procedures for finding optimal solutions in the presence of noise. It includes feedback iterations, precision checks and a restart procedure. We iterate between first-order and second-order approximations in order to continue the search for optima beyond the first-order approximation. We therefore also extensively discuss the use of stopping criteria.

Furthermore we study which settings and choices result in the best automated RSM procedure, both with regard to computing time and precision. In the literature, which is discussed extensively in section 2, we found rather confusing and non-systematic recommendations for the settings of such a procedure. In order to standardize the algorithm we fix some of the possible choices in the process based on the existing literature. Other choices are defined in a number of test algorithms that are compared using randomized deterministic test functions and a simulation model well known in the medical literature.

The setup of this paper is as follows. In section 2 we present our framework and we extensively discuss the choices to be made in the RSM procedure. We fix a number of the settings based on pre-tests and previous literature. Other choices are subject to experiments and the design of the experiments is discussed in section 3. The test functions and the

simulation model are described in section 4 and the results of our experiments are given in section 5. Finally, in section 6 we discuss the test results and give our recommendations.

2 Response Surface Methodology

In this section we will describe all steps in RSM with emphasis on the choices that need to be defined for an automated RSM algorithm. Without loss of generality, we discuss RSM for a minimization problem. Secondly, we assume that the objective function is of a stochastic nature, and that we want to optimize the expected value of the stochastic output. Mathematically, this problem can be written as

$$\min f : D \to \Re, D \subseteq \Re^k \tag{1}$$

where $f(\xi_1,...,\xi_k)$ is equal to $E(F((\xi_1,...,\xi_k))$. Here, $F((\xi_1,...,\xi_k)$ denotes stochastic output for given input $\{\xi_1,...,\xi_k\}$, and $E(F((\xi_1,...,\xi_k))$ denotes its expected value. We further assume that the variance in the function values is not known in advance. This situation especially arises in simulation studies, where the objective function can be seen as a black box that returns an output value for a given input. Simulation models do not assume a functional form and are subject to an unknown stochastic error. A simulation optimization exercise aims to find the input parameters that result into the minimum output value, such as finding the order-up to level that minimizes total costs in an inventory model.

We need to make some comments about above model and its assumptions. Firstly, note that the decision variables in problem (1) are continuous. Secondly, we consider an unconstrained optimization problem. Although we focus on this kind of problems, RSM can also be applied to constrained optimization problems. The interested reader is referred to Smith (1976), Myers and Montgomery (1995) and Angün (2004). Thirdly, since the simulation model is treated as a black box, variance reduction is only possible by applying more simulation runs.

In general the RSM procedure comprises two phases. In the first phase the stochastic objective function is locally approximated by a first-order polynomial and in the second phase by a second-order polynomial. For this purpose in both phases a region of interest (ROI) is defined, which is a sub-region of the domain. To obtain approximations of the stochastic objective function it is evaluated in the points of an experimental design. These points are usually located on the borders of the region of interest. When the first-order model is found to be adequate a steepest descent procedure is applied to find a new region of interest. Otherwise the RSM algorithm moves to the second phase. When a second-order model approximates the objective adequately, a stationary point is determined and classified and then an appropriate action has to be taken. Usually the algorithm is terminated and the stationary point is returned. However, we will discuss why it is profitable to continue the algorithm beyond the second phase especially for a stochastic function with unknown error variance.

In particular we will define an extension where we return to a first-order approximation in some cases and we include stopping rules based on the quality of the current centre point rather than a certain phase in the optimization process. Furthermore, we also propose to apply a restart procedure after the algorithm is ended for the first time. Figure 1 shows the optimization process on a global scale and it displays when a stopping rule is checked. The dotted arrows show the proposed extensions. The stopping rules applied are the same after first and second phase. In this perspective we define an iteration of the RSM algorithm as the run between two checks of the stopping criteria of the algorithm.

<< Insert Figure 1 about here.>>

There are a number of settings that need to be implemented in an automated RSM procedure using a consistent decision rule. These settings can be divided into what we call building blocks, strategic choices, stopping rules and a restart mechanism. In order to arrive at an automated RSM procedure we therefore follow to a large part the steps of the framework proposed by Neddermeijer et al. (2000a). Each step in this framework can be defined as a building block or a strategic choice. The building blocks of the algorithm consist of well-defined procedures that can be used to determine the next move of the algorithm. Strategic choices of the algorithm determine the action taken when a building block returns a result. In the next two sections we will discuss the literature and our extensions of the RSM algorithm by describing the building blocks and the strategic choices. Moreover, in sections 2.3 and 2.4 we define stopping rules and a restart mechanism. In section 2.5 we discuss a number of parameter settings, such as the significance levels of statistical tests that need to be chosen. Ultimately the description in the next sections leads to the framework shown in Figure 2.

<<Insert Figure 2 about here.>>

2.1 Building blocks

In this section we describe the building blocks applied in our procedure. In Figure 2 building blocks are given in rectangles.

2.1.1 Initialization

At the start of the algorithm an initial starting point and the initial size of the region of interest (ROI) should be given. A restart may ask for a different initialization: the starting point may be chosen randomly, be equal to the starting point used in the previous optimization run or equal to the best point in the previous optimization run. Beforehand it has to be specified when which choice is made. Note that this building block is not part of the first phase.

2.1.2 Approximate the objective function by a first-order model

In order to approximate the objective function it needs to be evaluated in the points of an experimental design. There are many designs available, like fractional or full factorial, and two-level or three-level designs (Myers and Montgomery, 1995). All designs can be augmented by the centre point of the region of interest. In non-automated optimization the user tries to fit a first-order approximation with different designs, apply coding of the factors to obtain better parameter estimates or recalculate the objective values in the design points. For instance, replicating the evaluation of the objective function in the centre point provides protection against curvature (Myers and Montgomery, 1995).

For an automated RSM procedure we follow the literature and evaluate the objective function once in the 2^k points of a two-level full factorial design and 5 times in the centre point of the current region of interest (Myers and Montgomery, 1995; Joshi, Sherali and Tew, 1998). This design is orthogonal and does not require as many points as a three-level full factorial design. In our opinion two-level fractional factorial designs consist of too few points to approximate objective functions with two or three parameters well enough. Furthermore, full factorial designs can quite easily be augmented to derive a second-order design (Neddermeijer et al., 2000a). Eventually, the coefficients of the first-order model are determined by applying least squares on the observed function values, while using coded variables to reduce the covariances of the parameter estimates.

2.1.3 Test the first-order model for adequacy

Usually, a test for lack of fit (Weisberg, 1985) and a test for significance of regression are performed (Myers and Montgomery, 1995). Box and Draper (1987) showed that the test for lack of fit is a joint test for interaction between factors as well as curvature. In non-automated optimization one can decide to use other tests and one can vary the significance levels based on the results from these tests. In automated optimization the levels should be fixed. This is subject to tests.

2.1.4 Perform a line search in the steepest descent direction

If the first-order model is found to be adequate a line search is performed from the centre point of the current region of interest in the steepest descent direction to find a point of improved response. Numerous implementations of the line search have been proposed (Box and Draper, 1987; Myers and Montgomery, 1995; Khuri and Cornell, 1996; Joshi, Sherali and Tew, 1998; Neddermeijer et al., 1999; Kleijnen et al., 2003). We will use increments $\Delta_1,...,\Delta_k$ along the path of steepest descent equal to the distance from the centre point to the point of intersection of the direction of steepest descent and the sphere given by $\sum_{i=1}^{k}\Delta_i^2 = 1$ (Neddermeijer et al., 1999). In a manual RSM algorithm one can observe the results of a line search and use personal likings to stop the search. However, in automated optimization, the algorithm needs a stopping rule that recognizes the lack of improvement in response during the line search.

The most straightforward rule ends the line search when an observed value of the objective function is higher than the preceding observation (Del Castillo, 1997). We will not use this stopping rule, known as the 1-in-a-row rule, because it is very sensitive to the noise in the response function function. In a similar way, the n-in-a-row stopping rule ends the line search when n observed values of the objective function are higher than the preceding observation. In the Myers and Khuri stopping rule (1979), the line search is ended when the mean response in a line search point, is significantly (statistically) higher than the mean response in a preceding line search point. This rule requires evaluating the response function function in a line search point more than once, because the variance of the response is not known at the start of the algorithm. In our algorithms we use the small sample t test in order to compare the mean responses in different points. This statistical test is robust with respect to both non-normality and unequal variances (Wackerley et al., 1996). We will test our version of the Myers and Khuri rule against the 3-in-a-row rule for our setting, where the variance in the stochastic objective function is not known a priori (contrary to the setup by Del Castillo and Myers & Khuri).

2.1.5 Approximate the objective function by a second-order model

The coefficients of the second-order model are again determined by regression analysis, applied to observations performed in an experimental design. A popular second-order design is the central composite design (CCD; Myers and Montgomery, 1995). The CCD arises when the full factorial design is augmented by adding $2k$ axial points (Box and Wilson, 1951). We make this design spherical by choosing the axial points such that all points are equidistant from the centre point of the current ROI **Error! Reference source not found.**. This design is near-rotatable **Error! Reference source not found.**. A fully rotatable design ensures equal variance of the estimate of the mean response at points equidistant from the centre point. However, for a fully rotatable design the distance of the axial points to the centre point would be large as compared to the distance of the existing points to the centre point.

2.1.6 Test the second-order model for adequacy

This module checks if the second-order model describes the behaviour of the objective function in the current region of interest. Similar to the first-order model a lack of fit test is performed. The null hypothesis of this test is that the true regression model is quadratic. In manual optimization one can use different significance levels or decide to overrule the outcomes of the lack of fit test. In automated optimization one has to determine these levels beforehand. The exaxt setting of the significance level is subject to tests.

2.1.7 Perform canonical analysis and, if necessary, ridge analysis

In the canonical analysis the stationary point of the second-order model is located and classified. Ridge analysis is done when the stationary point is a maximum, a saddle point or a minimum outside the current region of interest. In the ridge analysis we look for a point of minimum response inside the current region of interest since it is not correct to extrapolate the second-order model outside the current region of interest (Myers and Montgomery, 1995).

2.2 Strategic choices

In this section we describe the strategic choices that have to be made in our procedure. In Figure 2 these choices are given in ellipses. Although the choice whether to continue, to restart or to stop the algorithm is also a strategic choice, we discuss stopping rules and a restart mechanism separately in sections 2.3 and 2.4.

2.2.1 What to do when the first-order model is adequate?

If the first-order approximation is found to be adequate, a steepest descent procedure will be applied from the centre of the current ROI to find a new centre point (Box and Wilson, 1951; Box and Draper, 1987; Fu, 1994; Myers and Montgomery, 1995; Khuri and Cornell, 1996; Joshi, Sherali and Tew, 1998). This new point is then used as the centre point of the next region of interest. On this new region, the objective function will be approximated again by a first-order model (Myers and Montgomery, 1995).

2.2.2 How to solve first-order model inadequacy?

If the first-order model is not accepted, there is some evidence of curvature or interaction between factors on the current ROI, or the regression coefficients are all equal to zero. Most references suggest approximating the response function function by a second-order model (e.g. Fu, 1994; Myers and Montgomery, 1995; Neddermeijer et al., 2000a). An alternative is to increase the precision of the function evaluation in the design points. However this alternative is time-consuming and does not guarantee that the inadequacy is solved. We proceed with the second phase or our algorithm if the first-order model is inadequate.

2.2.3 What to do when the second-order model is adequate / How to proceed after the canonical analysis?

If the second-order approximation is found to be adequate then the appropriate action depends on the location and the nature of the stationary point. It is shown (Greenwood, Rees and Siochi, 1998) that for many functions a first-order model is inappropriate over a large percentage of the domain, so the algorithm can turn to the second phase quite early. The first stationary point found by a second-order approximation is therefore not likely to be the best point in the domain. If we stop the algorithm at this point (Fu, 1994; Kleijnen, 1998) the optimum could still be located far away from the current region of interest. We consider the following alternatives.

If a minimum is found inside the region of interest, this point will be used as the centre point of a new design and a new second-order approximation will be performed. We suggest to reduce the size of the ROI and to continue with phase 2 since we assume that we are close to the minimum of the objective function.

If the stationary point of the second-order polynomial is not a minimum inside the region of interest we perform ridge analysis to find a new stationary point (see section 2.1.7). We conclude that we are not close to the optimum and return to phase 1.

By considering these alternatives we now need stopping rules to decide when we are satisfied with the current solution. In section 2.3 we will discuss a number of stopping criteria and our tests will show the best stopping rules for specific functional forms.

2.2.4 How to solve second-order model inadequacy?

If the second-order model is inadequate, we conclude that either the region of interest is too large or that the stochastic nature of the function disturbs the approximation process. Stopping the algorithm at this point is only a good idea if there is an indication that the current centre point is close to the optimum. If we do not have any indication we propose to continue the algorithm and to redo the second phase. One way of solving the inadequacy of the approximation is increasing the precision used in evaluating a design point, i.e. variance reduction of the estimated response. This way the second-order polynomial will fit the objective function better. We could also reduce the size of the current region of interest like Joshi, Sherali and Tew **Error! Reference source not found.** propose. In our algorithm we will either reduce the size of the current region of interest or we will increase the precision used in evaluating the points of the second-order design. This is subject to tests.

2.3 Stopping rules

In automated optimization the RSM algorithm needs to be ended by consistent stopping rules that do not end the algorithm before a good solution is found and also do not unnecessarily prolong the algorithm. In section 2.2 we referred to the RSM literature where the optimization is ended after estimating only one second-order model (see **Error! Reference source not found.**,**Error! Reference source not found.**). We recommend ending the optimization if either the estimated response value does not improve sufficiently anymore, or, in case there are budget constraints, if a fixed maximum number of (function) evaluations has been performed. In this section we explain why these criteria seem to be consistent and how we apply them on the automatic algorithm. In our experiments we also consider the following stopping criterion: the algorithm is ended if the input values do not change suffiently anymore, i.e. if consecutive centre points are close to each other.

2.3.1 The estimated response does not improve sufficiently anymore (IMPROVE)

Algorithms for finding the optimum of a deterministic function can simply be ended when the function value does not improve sufficiently in consecutive iterations. When optimizing stochastic objective functions though, one has to take noise into account. Because we assume that the variance of the response is not known at the start of the algorithm we have to estimate it by evaluating the response in the new centre point of the region of interest more than once. We then need some statistical test to determine if there is sufficient progress or if different mean responses in two centre points are completely due to the noise. Notice that if the mean response does not decrease significantly in consecutive centre points we could still make progress. For instance the mean response may decrease from value 10.2 to 8.1 in 5 iterations, while in each separate iteration the change is not significant. We therefore implement the following criterion. Stop the algorithm if the mean response in the previous centre point does

not differ significantly from the mean response in the penultimate centre point in n consecutive iterations. It is important to note that the penultimate centre point is only changed in case the mean response differs significantly from the mean response in the previous centre point. The number of iterations is subject to tests. Notice that this stopping rule is based on the stopping rules for the steepest descent line search. It makes use of elements of both the Myers and Khuri rule as well as the n-in-a-row rule.

2.3.2 Convergence of input values (CONVERGE)

Algorithms that are used to find the optimum of a deterministic function are usually ended if the input parameters of the function do not change anymore. Therefore, we propose to end the algorithm if the Euclidian distance between two consecutive centre points is small, i.e. less than $\varepsilon\sqrt{k}$, where ε is a small number and k is the number of parameters of the objective function. In this way, the precision of each estimated parameter will be approximately equal to ε.

2.3.3 Fixed maximum number of function evaluations (MAXEVAL)

Our interest in the RSM is especially intended for stochastic models where the evaluation of the corresponding stochastic objective function is expensive or time-consuming. Therefore, ending the algorithm after a maximum number of function evaluations is appropriate when there are budget constraints. Notice that this stopping criterion does not consider the noise in the objective function.

The stopping rules discussed here will be applied after the first phase and after the second phase. It is then decided whether the algorithm is continued, restarted or really terminated.

2.4 A restart mechanism

Because RSM is a local search method there is no guarantee for finding a global optimum. The first centre point used in the RSM procedure is either selected by the user or randomly chosen and can influence the outcome of the procedure. Neddermeijer et al. (2000a) consider multiple starting points and/or multiple searches from the same starting point, when optimizing a stochastic objective function. In this study, we will use an adjusted mechanism that is based on these suggestions.

In order to escape from a non-optimal region we propose to restart the algorithm as soon as the algorithm is ended by one of the stopping criteria. The starting point of the restart is the best centre point of the 'normal run'. Because the algorithm cannot escape from a non-optimal region when the size of the region of interest is too small, we propose to reset the size of the region to its initial value. This way we ask the algorithm either to confirm the quality of the solution already found or to admit that there exists a better solution.

In general one should apply randomly chosen starting points when optimizing stochastic objective functions. If one has information about the unimodality of a function, it may not be necessary, especially if one has some idea about a good starting point. In the test problems we consider in this paper such was the case and we restart always in the same point. In this study we will run algorithms more than once starting from the same point. This can be seen as another restart mechanism and will give insight in the consistency of algorithms. We will come back to this in section 5.

In all optimization runs the best solution, i.e. the parameter values for which the mean response of the stochastic objective function is best, will be remembered. So our restart procedure cannot deteriorate the solution found in the 'normal' optimization run. Note that since the response function is stochastic, it is possible that the mean response measured in a non-optimal point is better than the mean response measured in the real optimal point.

2.5 Parameter settings

A number of parameter settings will be subject to pre-tests. For instance, the lack of fit test can be performed at significance levels of 2.5%, 5% or 10%. The size of the increment in the precision of the function evaluation in the design points is also subject to tests. Furthermore, it is tested by how much the size of the region of interest is decreased in order to solve the inadequacy of the second-order model.

3 Test design

In this section we discuss our test design. In a pre-testing phase we have determined the parameter values that give the best performance with respect to running time and quality of the solutions returned. In our experiments we consider only choices that we feel have the most impact on the efficiency of the RSM procedure. We benchmark our setup against the setup by Fu (1994) who stops after doing the second phase only once.

In particular we consider the following alternatives. We test the 3-in-a-row stopping rule against our version of the Myers and Khuri stopping rule for the steepest descent line search. Myers and Khuri (1979) conclude that their stopping rule dominates the 3-in-a-row rule for stochastic functions with known variance. When the variance is not known we are uncertain if the Myers and Khuri (M&K) rule is still as successful. We expect that algorithms using our version of the M&K rule are less efficient since every point in the line search is evaluated 5 times.

When the second-order model is found to be adequate and a minimum is found within the region of interest we assume that we are close to an optimal solution. We then shrink the region of interest with either 50% or 10%. These values are already set by pre-testing, but the difference between these shrink-percentages is important. When the size of the region of interest is decreased by 10% it will take more time to focus on a certain region that is suspected to contain the global optimum and the algorithm could be prolonged unnecessarily. However if we shrink the region of interest too soon, we are at higher risk of returning a non-optimal solution. The success of the alternatives will closely interact with the stopping rules and the restart procedure. For instance, we expect that the 50% shrink procedure (ensures fast convergence) in connection with a strict stopping rule with respect to noise (ensures fast convergence) and a restart procedure that returns to the original size of the region of interest (recognize quality of solutions) will perform well.

When the second-order approximation is rejected either the noise is dominant or the region of interest is too large. In order to solve this problem we either reduce the noise, e.g. by performing more simulation runs, or we reduce the size of the region of interest. We think that algorithms using noise reduction will give more accurate results since the quality of the estimates of the response function is increased. We expect the running time to be higher for these algorithms.

As a result we will test eight algorithms, using all proposed alternatives in combination with the others. For all these eight algorithms we will study the effect of the stopping rules and restart procedure explained in sections 2.3 and 2.4.

<< Insert Table 1 about here.>>

The setup of the experiments is as follows. First we apply the algorithms for a large number of iterations and for each algorithm we record the relevant values for the application of the stopping criteria. Especially, at each iteration we record (the mean response in) the

current (best) centre point, the number of function evaluations and the Euclidean distance between the current and the last centre point. We then decide on the exact setting of each stopping rule such that the average performance of each algorithm for a test set defined in the next section is "best". Each algorithm can have a different setting of the stopping rules for which it performs best.

In the first place, the criteria for the preferred performance of the stopping rules are based on the quality (or precision) of the solutions compared to the deterministic values of the test functions. Precision is measured in two ways. On the one hand we look at the error of an optimization run, which is defined as the difference between the expected simulation response function value in the true optimal point and the expected simulation response in the observed best point of the run. On the other hand we consider the distance of an optimization run, which is defined as the Euclidean distance between the true optimal point and the observed best point of the run. Notice that we have full knowledge of the optima of the test functions and corresponding solution values. We can therefore observe the parameters of each stopping rule of the RSM procedure for every test function and determine when the procedure is no longer effectively improving the solutions.

Secondly, the preferred performance of the stopping rule is also based on the running times of the algorithms needed to fulfil the different settings of the stopping criteria. We measure the running time by the number of function evaluations. Note that the standard deviation of the error and the distance of optimization runs can be used as an indicator of the consistency of an algorithm.

The settings of the algorithms are now taken such that the average performance, i.e. quality and running time, over the test functions is best. We do this because we assume that we have no previous knowledge of the objective function. Note that we apply each algorithm on each test function 100 times to find the average behaviour of the algorithm.

In a second phase of experiments we run the algorithms again for the new settings of the stopping rules and we apply a restart procedure when each algorithm is stopped. The restart procedure is applied once and the result of the algorithm after the restart is returned as the output of the algorithm. We apply each algorithm 100 times on each test function. Note that the replication of an optimization run is in itself a restart procedure and therefore allows us to compare two different restart strategies. The performance of the eight algorithms is compared to the performance of four algorithms using the setup of Fu (1994), applied to the same set of test functions. Note that we deal with four algorithms since these algorithms are ended when the objective is approximated adequately by a second-order polynomial once. So, for these algorithms the third column of Table 1 ('Shrink region of interest') is of no importance. We will speak about these algorithms as the Fu versions of the algorithms in Table 1.

4 The Test Problems

We will test the optimization algorithms on a set of seven test functions, which consist of a deterministic and a stochastic term. Their optima are known. These functions were also used in comparing different versions of Nelder and Mead simplex method (NMSM) (Neddermeijer et al., 2000b) and in comparing RSM algorithms with NMSM algorithms (Neddermeijer et al. 1999). We will also consider a simulation version of an existing cancer-screening model. This model has three parameters that need to be estimated from an observed data set using minimization of a goodness-of-fit statistic (Neddermeijer et al., 1999). For this particular model the optimal parameters can also be determined analytically, but we are interested in the performance of the RSM procedure for this simulation exercise.

4.1 One-stage-one test breast cancer model

The simulation 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 pre-clinical phase (DPCP) is modelled. The DPCP has incidence rate J and we assume that the duration of the DPCP is exponentially distributed with parameter λ. 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 program of four annual screening rounds is simulated. The sensitivity of the screening test is denoted by φ. In each simulation run 50,000 individual life histories, including the disease processes and the impact of screening, are simulated. The simulation model generates 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 consist of 4 detection rates and 14 incidence rates of interval cancers occurring after a previous negative test result.

The parameters J, λ, and φ will be estimated from the observed data set through minimization of a chi-square goodness-of-fit test statistic. The simulation objective function is given by

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

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,..,18.

The true optimal parameters of the model for the HIP data were derived using the objective function

$$f(J,\lambda,\varphi) = \sum_{i=1}^{18} (O_i(J,\lambda,\varphi) - A_i(J,\lambda,\varphi))^2 \,/\, A_i(J,\lambda,\varphi) \tag{3}$$

where A_i is the number of cancers during screening round or interval i, i=1,..,18, as predicted by the analytical implementation of the breast cancer screening model (Day and Walter, 1984). The optimal parameters (J^*, λ^*, φ^*) of the model, applied to the HIP data, are determined by extensive enumeration (using step sizes 10^{-5} for J, 10^{-3} for λ, and 10^{-3} for φ) of $f(J,\lambda,\varphi)$: $f(J^*,\lambda^*,\varphi^*) = f(0.00213, 0.614, 0.871) \approx 13.343$ (Neddermeijer et al., 2000b).

4.2 The test functions

In addition to the HIP screening simulation model we test randomized versions of seven deterministic unconstrained nonlinear minimization problems. All problems have a unique global optimum. We think that the test functions show a characteristic behaviour that may

occur in stochastic objective functions resulting from simulation models. We randomize the deterministic test functions by adding a normal distributed error term with zero mean and variance $\sigma^2 = 1$. For each optimization run we use independent random number streams. The test functions are:

1. Rosenbrock function

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

The minimum of this classical test function, given by $f(1,1) = 0$, lies at the base of a banana-shaped valley (Gill, Murray and Wright, 1981, among many others). Nearby this minimum, the function increases fast in the direction of the second parameter.

2. Powell singular function

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

This function has four parameters. The minimum is given by $f(0,0,0,0) = 0$. Apart from a small region around the optimum this function is very steep.

3. A 5-variable parabolic function

$$f(x_1, .., x_5) = \sum_{i=1}^{5} x_i^2$$

This function is symmetrical and rather easy to optimize if no noise is included (Neddermeijer et al., 2000b). The minimum is given by $f(0,0,0,0,0) = 0$.

4. Symmetrical Gaussian function

$$f(x_1, x_2) = \text{-}10\exp\{\text{-}[(100 - x_1)^2 + (100 - x_2)^2]/15000\}$$

Apart from a small region around the optimum this symmetrical test function is very flat (Van der Wiel, 1980; Neddermeijer et al., 2000b). The minimum of this function is given by $f(100,100) = -10$.

5. An asymmetric function

$$f(x_1, .., x_8) = \sum_{i=1}^{8} [2^{x_i - 4} + (6 - x_i)]$$

Just like the simulation model, this test function is asymmetric. The minimum is given by $f(4.529,..,4.529) = 23.311$.

6. Beale function

$$f(x_1, x_2) = \left[1.5 - x_1(1 - x_2)\right]^2 + \left[2.25 - x_1(1 - x_2^2)\right]^2 + \left[2.625 - x_1(1 - x_2^3)\right]^2$$

This function is very steep apart from a small region around the optimum. Nearby this optimum, given by $f(3, 0.5) = 0$, the function increases relatively fast in the direction of the second parameter.

7. Wood function

$$f(x_1, x_2, x_3, x_4) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2$$

$$+10.1((1-x_2^2)^2+(1-x_4)^2)+19.8(1-x_2)(1-x_4)$$

This function is very steep. The minimum is given by $f(1,1,1,1)=0$.

5 Results

Below we discuss the results of the experiments. From preliminary tests we obtained the best settings of the parameters we described in section 2.5. Table 2 shows these settings.

<< Insert Table 2 about here.>>

The last row of Table 2 contains two entries: 0.95 and 25%. For the seven randomized deterministic test problems we multiply the variance by 0.95; for the simulation model we increase the number of simulated life histories by 25% in order to solve second-order model inadequacy. We expect that the latter has more or less the same effect on the variance of the objective function of the simulation model.

5.1 Phase 1

In phase 1 of our numerical experiments we determine the exact settings of the stopping criteria. To this end we have recorded the results of the eight specified algorithms for a number of settings of the stopping criteria. These settings are shown in Table 3. Note that the settings of the stopping rules may depend on the number of parameters (k) of a test problem.

<< Insert Table 3 about here.>>

In Tables 4a-c the results of the application of algorithm 1 on test problems 2, 6 and 7 are shown. The first column in each Table contains the description of the stopping criteria and in the second column the setting of each criterion is given. The third column (ERROR) shows the mean (standard deviation) of the absolute differences between the optimal value and the expected value (i.e. without noise) of the objective function in the solution found. In the fourth column (DISTANCE) the mean (standard deviation) of the Euclidean distances between the actual minimum and the solution found is given. The fifth column (nEVAL) shows the mean (standard deviation) of the number of evaluations needed to fulfil the different stopping criteria. The last row contains the results of ending the algorithm after the maximum number of iterations, in this case 150.

<< Insert Tables 4a-c about here.>>

The results in Tables 4a-c show that the more iterations an algorithm runs the lower the error is. However, it appears that the Euclidean distance between the estimated parameters and the actual location of the minimum is not always smaller as the algorithm runs longer. This especially holds for the optimization of test problems 2 and 6, i.e. the Powell and the Beale function. This is probably caused by the fact that these functions are very steep. We expect that the restart mechanism solves this problem by enlarging the ROI.

It is also striking that in general the standard deviations of the errors of the algorithms are roughly of the same order of magnitude as the errors themselves. This points out that the algorithms do not consistently perform well; a small part of the solutions found worsens the

mean error. However, there is a high probability that a good solution is found when we apply an algorithm more than once on the same problem.

As it was stated before, the performance of an algorithm does not only depend on the precision of the solutions found, but also on the computing time. When optimizing stochastic objective functions the number of function evaluations needed to obtain a certain solution is an important indicator for the computing time. The values in the last column in Tables 4a-c show that the difference in the number of function evaluations between the least restrictive setting and the most restrictive setting of stopping criteria CONVERGE and IMPROVE can be quite large. The extra running time needed to increase the precision does not always balance the extra computing time. For example, for test function 6 we find that the second setting of CONVERGE requires 569 evaluations on average. The precision of the solution found however, is not much worse than the solution resulting from ending the algorithm by the fourth setting of CONVERGE, requiring twice as many evaluations. Moreover, when the algorithm is ended after 150 iterations, requiring more than 3000 evaluations, the precision of the solution found is not much better. It only seems useful to prolong the optimization run for test problems 2 and 7 which are very steep functions.

The mean error, the mean Euclidean distance and the mean number of evaluations do not tell the whole story. Therefore we have also compared the 100 observed errors, distances and number of evaluations separately for all algorithms by using nonparametric statistical tests. This way we determined whether there is any stochastic difference between the different settings of the stopping criteria CONVERGE, IMPROVE and MAXEVAL. The specific nonparametric test we applied is the Kruskal-Wallis test (see e.g. Wackerley et al. (1996)).

The nonparametric tests do not give identical answers for the different test problems. It appears that for most of the test functions less restrictive settings of the three stopping criteria do not result in statistically lower errors and distances. That is, running the algorithms longer does not improve the precision of the solutions found. However, the errors and distances for the Wood and the Powell function and to a less extent the Beale and the Rosenbrock function are significantly lower when the settings are less restrictive. Of course the computing time, measured as the number of function evaluations, increases significantly when the settings of the stopping criteria are less restrictive. The results of the nonparametric tests are uniform over the different algorithms.

Since we want to find algorithms that optimize all test functions very precisely we cannot choose the settings of the stopping rules too restrictive (in phase 2). In Table 5 the settings of the rules IMPROVE and CONVERGE are given. These settings (see Table 3 for their definition) are used to end the algorithms in our second phase of experiments. We decided not to use MAXEVAL since this rule ends the algorithms on almost the same moment as IMPROVE. We prefer IMPROVE since it takes noise into account whereas MAXEVAL is a more rigid rule. As a consequence of this choice, we allow both IMPROVE and CONVERGE to terminate the algorithms in the second phase of experiments.

<< Insert Table 5 about here.>>

5.2 Phase 2

In phase 2 of the experiments we do not only consider the eight algorithms using the settings as given in Table 5. We also test versions of these algorithms that are ended quite early. We expect that a comparison of both groups of algorithms gives us some understanding of the performance of our algorithms and in particular of the quality of our stopping rules. Moreover we test whether restarting the algorithm improves the precision of the solution.

In Tables 6a-6d the results of applying the four Fu versions of the algorithms are shown. Note again that we use the numbering of the algorithms as given in Table 1. This is done since the four Fu versions are copies of the algorithms in Table 1, but they are ended when an adequate second-order model is estimated. In these algorithms there is no action taken when a minimum is found. So, we have applied Fu versions of algorithms 1, 2, 5 and 6. In the first column of Tables 6a-6d the name of each test function is shown. In the second column the mean (standard deviation) of the absolute deviations between the actual objective value in the optimum and the actual value of the objective function in the minimum found (i.e. the error) is given. The third column shows the mean number of function evaluations that the algorithm has carried out to find the minimum. The fourth and fifth column contain the mean (standard deviation) of respectively the errors and the additional number of evaluations when the restart mechanism is also applied.

<< Insert Tables 6a-6d about here.>>

Tables 6a-6d show that for the most test functions the restart mechanism really improves the precision of the solution found after the 'normal' run. The effect is not that large for the simulation model, but in this case the restart contributes to more consistent performance of the algorithm. Actually, for this particular test problem the normal run already finds quite accurate solutions. The number of additional function evaluations done in the restart is relatively small. We have to say however that in general the quality of the solutions found in the normal run is quite bad. For example, if we compare the errors found in the normal run of the four algorithms with the errors found by algorithm 1 in phase 1 using the most restrictive setting of IMPROVE (see Tables 4a-4c), we conclude that the former errors are much higher than the latter. The number of evaluations needed when using the most restrictive setting of IMPROVE is in most cases much higher than it is for the Fu versions. This means that the Fu versions of the algorithms do not perform enough function evaluations to find an accurate solution.

In Table 7a the mean and standard deviation of the errors resulting from the application of the eight different algorithms on the test functions are shown. It can be seen that in general algorithm 5 finds the smallest errors and standard deviations. Note that algorithm 5 ends the line search using the Myers and Khuri rule, shrinks the region of interest by 50% when a minimum (inside this region) is found and solves the second-order model by reducing the noise in the objective values.

<< Insert Table 7a about here.>>

We also find that the precision and consistency of the eight algorithms is higher than it is for the Fu versions. This is also proved by nonparametric statistical tests. Furthermore, a nonparametric statistical test indicates that the distance between the actual optimal solution and the solution found by the eight algorithms is lower than it is for the Fu versions. On the other hand, the Fu versions need significantly less function evaluations to find solutions.

Although the eight algorithms perform more consistently than the Fu versions the standard deviations of the errors are still of the same order of magnitude as the errors. This means that we cannot claim that these algorithms are consistently performing well. Actually, it turns out that the algorithms are sometimes stuck in non-optimal points. It appears that in a number of optimization runs the first- and second-order models are repeatedly inadequate in the first few iterations. As a consequence the region of interest is decreased repeatedly and the algorithm is ended by one of the stopping rules. Of course, one or two of these bad solutions can really

influence the mean error. We have to note here that in the point (0,0) one of the partial derivatives of the Rosenbrock function is equal to 0. The objective value in this point is 1. The algorithms are usually stuck in this point, but this is not caused by inadequate approximations of the functional form of the objective.

In Table 7b the mean and standard deviation of the ten smallest errors resulting from the application of the eight different algorithms on the test functions are given. It appears that the mean and standard deviation of the ten smallest errors are small compared to the mean and standard deviation of all errors. This indicates that the algorithms perform very well in at least 10% of the optimization runs. On average, algorithms 3 and 5 find the best solutions. Note that the best solutions can be retrieved by running many simulations for the parameter solutions found by an algorithm. This way the best solutions can be selected in case this cannot be determined analytically. In conclusion, performing a number of optimization runs and then selecting the best solutions can be really advantageous. We come back to this later.

<< Insert Table 7b about here.>>

In Table 7c the mean and standard deviation of the number of evaluations resulting from the application of the eight different algorithms on the test functions are shown. It can be seen that algorithm 5 needs the highest number of evaluations, its natural counterpart, i.e. algorithm 4, needs the least number of evaluations. This is not surprising since algorithm 5 uses the Myers and Khuri rule to end the line search. After all, this rule requires evaluating the objective function more than once in each point on the line. It follows from Table 7c that algorithms using this rule in general need more evaluations. Algorithms using noise reduction to solve second-order model inadequacy also appear to evaluate the objective function more than algorithms that shrink the region of interest. Moreover, algorithms that shrink the region of interest by 50% when a minimum is found need more evaluations then algorithms that shrink it by 10%. We cannot explain the latter results; we expected the counterpart since shrinking the region of interest by 50% a number of times results in a very small region.

<< Insert Table 7c about here.>>

In Table 8a the mean and standard deviation of the errors resulting from the application of the eight different algorithms using the restart mechanism on the test functions are shown. It appears that the Powell and Wood function benefit the most from the restart. Both the mean as well as the standard deviation of the errors are lower for these functions after applying the restart. It can also be seen that in general algorithm 5 finds the smallest errors and standard deviations. However, the standard deviations of the errors of this algorithm are in 5 out of 8 cases of the same order of magnitude as the errors themselves. This means that bad solutions are still found.

<< Insert Table 8a about here.>>

In Table 8b the mean and standard deviation of the ten smallest errors resulting from the application of the eight different algorithms using the restart mechanism on the test functions are shown. Again we see that the ten best solutions found are much better than the other solutions. Moreover, these solutions are better than the solutions found in the original run of the algorithms. The mean errors of algorithms 3 and 5 are smallest.

<< Insert Table 8b about here.>>

In Table 8c the mean and standard deviation of the number of evaluations resulting from the application of the restart of the eight different algorithms on the test functions are given. The results show that algorithms 1 and 5 need the highest number of evaluations in the restart; algorithm 4 needs the least number of evaluations.

<< Insert Table 8c about here.>>

In Figures 3a-3h we have plotted the mean number of evaluations against the mean error of every algorithm for each test function. This way we can distinguish efficient algorithms, that is algorithms that find precise solutions in reasonable time. We call algorithms efficient when they are closer to the origin than other algorithms. Note that we did not plot all test functions in each Figure; if the error of an algorithm is too big the algorithm is not in the Figure. It appears that algorithm 5r (i.e. algorithm 5 using the restart mechanism) finds precise solutions for almost every test function, whereas algorithm 4 is the fastest algorithm. However, as we observed earlier, there is no particular algorithm that is really efficient, i.e. both fast and precise. It seems that algorithm 7 using the restart is most efficient among the algorithms we have studied concerning the average precision over 100 optimization runs. After all, this algorithm appears to be closest to the origin in most of the Figures. We also have to mention algorithm 5 since it is almost equally efficient for a number of test functions. Note that the only difference between algorithms 5 and 7 is the 'shrinking percentage'. Also note that the reader is allowed to determine for himself which algorithm is most efficient. The efficiency utility curve expressing how much value one assigns to a combination of precision and computing time, may differ among people. Some prefer fast algorithms with relatively low precision to relatively slow algorithms with high precision. Others may think precision and computing time are equally important.

<< Insert Figures 3a-3h about here.>>

Figures 4a-4h show plots of the mean number of evaluations against the mean error of the best ten solutions of every algorithm for each test function. Here we use the mean number of evaluations over all 100 optimization runs, since we need to do all runs before we can determine the best ten solutions. We mentioned earlier that the best ten solutions of algorithm 3 (without restart) are relatively precise. It appears that this algorithm does not need much more evaluations than algorithm 4, which is the fastest algorithm on average. So we may call algorithm 3 efficient among the other algorithms, in the sense that selecting the ten best solutions of this algorithm results in a high precision in reasonable time. Algorithm 3 uses the 3-in-a-row rule and noise reduction. The former setting yields a lower computing time, the latter a higher precision. Apparently this combination is efficient.

<< Insert Figures 4a-4h about here.>>

6 Discussion

In this section we will discuss a number of important issues when using an optimization technique like RSM. First of all, the algorithms we have studied can serve as a benchmark algorithm in other studies. Here we aim at algorithm 5 in combination with the restart mechanism, since it is the most precise algorithm, but also at algorithm 3, which seems to be the most efficient algorithm concerning the ten best solutions. Moreover, algorithm 7 can be used as benchmark for testing the efficiency concerning all solutions.

Secondly, although the restart mechanism should not be seen as a procedure that always recovers an algorithm when it is stuck in a bad point (solution), we have seen that the restart mechanism increases the precision of the algorithm for a number of test functions. The main lesson is that regularly enlarging the region of interest during the optimization run, may lead to better solutions. An important question which is unanswered is how can we employ time (read: the number of evaluations) best to search the domain of the objective function. One answer to this question is using strict stopping criteria and multiple restarts.

Thirdly, in this paper we have applied all algorithms 100 times to each optimization problem. We have seen that the algorithms do not perform consistently well. On the other hand determining the best ten of the 100 solutions shows that the algorithms also find very good solutions. Therefore it seems profitable to run an algorithm more than once and then select the best solutions. Selecting good solutions is not easy; Boesel et al. (2003) present an advanced method to select the best solution from all solutions visited during an optimization run. We feel that running an optimization algorithm 100 times and then re-evaluating the ten solutions with the lowest estimated response is a good way to find a good solution. Since we have an estimated response based on 5 function evaluations for every solution, we do not need additional evaluations for determining the best ten. Next, the ten best solutions are re-evaluated using more simulation runs than used in the optimization procedure. The resulting estimated responses are then compared using standard techniques.

Fourthly, since RSM uses factorial designs it can only be applied to problems with a small number of variables. Even when the objective function is evaluated in the points of a fractional two-level factorial design, the number of evaluations increases exponentially with the number of parameters. For instances with, say, more than ten parameters, using techniques like the Nelder and Mead Simplex method seem more appropriate. This method appears to perform quite well, concerning both precision and computing time, for bigger problem instances (see e.g. Barton and Ivey, 1996; Neddermeijer et al. 2000b).

7 Conclusions

In this paper we worked towards a standardized automated RSM algorithm. The basis for this algorithm is the framework of Neddermeijer et al. (2001). We have extended this by introducing stopping rules and a restart mechanism. Consequently, different settings for such an automated RSM procedure for simulation optimization are analyzed. We compared the precision and efficiency of the different algorithms for optimization of a set of test problems, including a simulation model for cancer screening. Below we summarize our findings.

- The RSM algorithms proposed in this paper are suitable for optimizing objective functions with up to eight variables quite precisely in reasonable time. Especially algorithm 3 appears to be efficient in this respect. The best ten solutions of this algorithm are of high quality; the number of evaluations needed to find these solutions is quite low.

- The test problems considered in this paper have unimodal objective functions, which means that they do not have local optima. Nevertheless it appears that the RSM algorithms get sometimes stuck in non-optimal solutions. Generally, this occurs when the objective function is very flat in comparison with the associated noise on a certain region of the domain. This problem can be solved effectively by using the restart mechanism. Resetting the region of interest to its initial size, i.e. an enlargement of this region, and starting the search all over again, enables the algorithm to escape from regions where the objective function is very flat.

- We have tested a number of refinements. It appears that some of the modified RSM procedures, e.g. the algorithms using noise reduction and the Myers & Khuri rule, find more precise solutions. Unfortunately, using these settings has a drawback: the computing time measured as the number of function evaluations, is higher for these algorithms.

- It appears that rerunning the algorithms can lead to very different results. The standard deviations of the error of the algorithms are sometimes, even when the restart mechanism is applied, of the same order of magnitude as the errors themselves. This means that there is no consistency in the quality of solutions. On the other hand, running an algorithm ten times already gives a very high probability of finding a good solution.

- Stopping rules indeed make that the RSM procedures recognize when no further improvement is being made. In the first phase of our experiments we noticed that most of the algorithms were terminated too early. This leads to low precision. Moreover, the Fu versions of the algorithms, which are ended after only one second-phase, are also terminated too early. However, it is a little disappointing that the eight tested algorithms need quite a lot of evaluations to decrease the mean error of all 100 optimization runs.

ACKNOWLEDGEMENTS

The authors would like to thank Nanda Piersma and Gerrit van Oortmarssen who initiated this research project. We also thank Alex Koning for his useful comments and suggestions regarding the statistical aspects of this paper.

References

Angün, E. (2004). Black box simulation optimization: Generalized Response Surface Methodology. PhD thesis, Tilburg University.

Barton, R.R., J.S. Ivey (1996). Nelder-Mead Simplex Modifications for Simulation Otimization. *Management Science* **42**(7) 954-973.

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

Box, G.E.P., and K.B. Wilson (1951). On the Experimental Attainment of Optimum Conditions. *Journal of the Royal Statistical Society* **B13,** 1-38, discussion: 38-45.

Boesel, J., B.L. Nelson, and S.-H. Kim (2003). Using Ranking and Selection to `Clean Up' After Simulation Optimization. *Operations Research* **51**, 814-825.

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

Del Castillo, E. (1997). Stopping Rules for Steepest Ascent in Experimental Optimization. *Communications in Statistics. Simulation and Computation* **26**(4), 1599-1615.

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

Gill. P.E., W. Murray, and M.H. Wright (1981). *Practical Optimization.* London: Academic Press.

Glover, F, J. Kelly and M. Laguna (1999). New advances for wedding optimization and simulation, *Proceedings of the 1999 Winter Simulation Conference*, P. Farrington, H. Nembhard, D. Surrock and G. Evans (Eds.), pp. 255-259. Piscataway, NJ: IEEE.

Greenwood. A.G., L.P. Rees and F.C. Siochi (1998). An Investigation of the Behavior of Simulation Response Surfaces. *European Journal of Operational Research* **110**, 282-313.

Joshi, S., H.D. Sherali and J.D. Tew (1998). An Enhanced Response Surface Methodology (RSM) Algorithm Using Gradient Deflection and Second-Order Search Strategies. *Computers and Operations Research* **25**(7/8), 531-541.

Khuri, A.I., and J.A. Cornell (1996). *Response Surfaces: Designs and Analyses.* New York: Marcel Dekker, Inc.

Kleijnen, J.P.C., 1998. Experimental Design for Sensitivity Analysis, Optimization, and Validation of Simulation models. In *Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice*, ed. J. Banks, 173-223. New York: John Wiley & Sons.

Kleijnen, J.P.C., D. den Hertog, and E. Angün (2004). Response surface methodology's steepest ascent and step size revisited, *European Journal of Operational Research* 159, 121-131.

Myers, R.H. and A.I. Khuri (1979). A New Procedure for Steepest Ascent. *Communications in Statistics. Theoretical Methods* **A8**(14), 1359-1376.

Myers, R.H., A.I. Khuri, and W.H. Carter, Jr. (1989). Response Surface Methodology: 1966-1988. Technometrics 31(2), 137-157.

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

Neddermeijer, H.G., G.J. van Oortmarssen, N. Piersma, and R. Dekker (2000a). A Framework for Response Surface Methodology for Simulation Optimization. In *Proceedings of the 2000 Winter Simulation Conference*, J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick (eds.), pp. 129-136.

Neddermeijer, H.G., G.J. van Oortmarssen, N. Piersma, R. Dekker and J.D.F. Habbema (2000b). *Adaptive Extensions of the Nelder and Mead Simplex Method for optimization of stochastic simulation models.* Econometric Institute Report EI2000-22/A. Erasmus Universiteit Rotterdam, The Netherlands.

Neddermeijer, H.G., N. Piersma, G.J. van Oortmarssen, J.D.F. Habbema and R. Dekker (1999). *Comparison of Response Surface Methodology and the Nelder and Mead Simplex Method for Optimization in Micro simulation Models.* Econometric Institute Report EI-9924/A. Erasmus Universiteit Rotterdam, The Netherlands.

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

Smith, D.E. (1976). Automatic Optimum-Seeking Program for Digital Simulation. *Simulation* **27**, 27-31.

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

Wackerley, Mendenhall and Scheaffer (1996). *Mathematical Statistics with Applications*. 5th Edition, Duxbury.

Weisberg, S. (1985). *Applied Linear Regression.* New York : John Wiley & Sons.

Wiel, P.F.A. van der (1980). Improvement of the Super-Modified Simplex Optimization Procedure. *Analytic Chimica Acta* **122**, 421-433.

Algorithm	Stopping rule steepest descent	Shrink region of interest	Solve second-order model inadequacy
1	3-in-a-row	50%	Noise reduction
2	3-in-a-row	50%	Shrink design
3	3-in-a-row	10%	Noise reduction
4	3-in-a-row	10%	Shrink design
5	Myers and Khuri	50%	Noise reduction
6	Myers and Khuri	50%	Shrink design
7	Myers and Khuri	10%	Noise reduction
8	Myers and Khuri	10%	Shrink design

Table 1: The test design

Parameter	Setting
Significance level of all statistical tests	5%
Shrink design for solving second-order inadequacy	50%
Increase precision for solving second-order inadequacy	0.95 / 25%

Table 2: Parameter settings

Criterion	Setting
IMPROVE	$5 \cdot j, \quad j = 1,2,3,4,5$
CONVERGE	$5E\text{-}04 \cdot j \cdot \sqrt{k}, \quad j = 1,2,4,10,20$
MAXEVAL	$50 \cdot j \cdot 2^k, \quad j = 1,2,3,4,5$

Table 3: Possible settings of stopping criteria

Criterion	SETTING	ERROR	DISTANCE	nEVAL
IMPROVE	5	0.746 (0.60)	0.288 (0.086)	1628 (238)
	10	0.472 (0.41)	0.263 (0.075)	2113 (525)
	15	0.370 (0.37)	0.259 (0.080)	2735 (1253)
	20	0.302 (0.31)	0.260 (0.090)	3601 (1976)
	25	0.261 (0.27)	0.259 (0.091)	4562 (2686)
CONVERGE	1.0E-03	0.276 (0.35)	0.260 (0.092)	5542 (3061)
	2.0E-03	0.208 (0.27)	0.265 (0.091)	8864 (3707)
	4.0E-03	0.172 (0.15)	0.262 (0.078)	11634 (3807)
	1.0E-02	0.187 (0.16)	0.264 (0.078)	12898 (3258)
	2.0E-02	0.185 (0.14)	0.267 (0.095)	13598 (2341)
MAXEVAL	800	3731 (642)	3.94 (0.33)	800 (0)
	1600	0.665 (0.60)	0.280 (0.085)	1600 (0)
	2400	0.369 (0.38)	0.255 (0.081)	2400 (0)
	3200	0.278 (0.27)	0.254 (0.079)	3200 (0)
	4000	0.260 (0.27)	0.256 (0.081)	4000 (0)
Max. iterations	500	0.169 (0.17)	0.278 (0.11)	18103 (236)

Table 4a: Results of applying algorithm 1 on test problem 2

Criterion	SETTING	ERROR	DISTANCE	nEVAL
IMPROVE	5	0.329 (0.31)	1.88 (0.029)	281 (118)
	10	0.283 (0.19)	1.87 (0.035)	517 (269)
	15	0.266 (0.18)	1.88 (0.036)	805 (405)
	20	0.255 (0.17)	1.88 (0.045)	1038 (520)
	25	0.254 (0.17)	1.88 (0.044)	1218 (613)
CONVERGE	7.071E-04	0.319 (0.30)	1.88 (0.038)	354 (193)
	1.414E-03	0.264 (0.19)	1.88 (0.046)	569 (305)
	2.828E-03	0.249 (0.17)	1.88 (0.052)	863 (418)
	7.071E-03	0.241 (0.16)	1.88 (0.054)	1076 (458)
	1.414E-02	0.243 (0.16)	1.88 (0.054)	1266 (542)
MAXEVAL	200	0.347 (0.32)	1.88 (0.024)	200 (0)
	400	0.275 (0.20)	1.87 (0.035)	400 (0)
	600	0.263 (0.18)	1.87 (0.038)	600 (0)
	800	0.262 (0.18)	1.88 (0.042)	800 (0)
	1000	0.241 (0.16)	1.88 (0.056)	1000 (0)
Max. iterations	150	0.228 (0.14)	1.88 (0.074)	3272 (133)

Table 4b: Results of applying algorithm 1 on test problem 6

Criterion	SETTING	ERROR	DISTANCE	nEVAL
IMPROVE	5	4.96 (4.0)	0.533 (0.19)	1720 (402)
	10	2.76 (3.7)	0.396 (0.20)	2577 (735)
	15	1.99 (2.6)	0.348 (0.17)	3515 (1247)
	20	1.62 (2.1)	0.325 (0.16)	4588 (1941)
	25	1.50 (2.1)	0.315 (0.16)	5775 (2598)
CONVERGE	1.0E-03	16.7 (14)	0.930 (0.36)	1128 (586)
	2.0E-03	2.42 (4.0)	0.360 (0.22)	3507 (1724)
	4.0E-03	1.26 (2.0)	0.294 (0.16)	9183 (3780)
	1.0E-02	1.38 (2.0)	0.310 (0.16)	11109 (4137)
	2.0E-02	1.57 (2.2)	0.323 (0.17)	11366 (3945)
MAXEVAL	800	31.8 (25)	1.27 (0.45)	800 (0)
	1600	5.62 (4.0)	0.572 (0.18)	1600 (0)
	2400	2.64 (3.3)	0.395 (0.18)	2400 (0)
	3200	2.04 (2.9)	0.351 (0.18)	3200 (0)
	4000	1.67 (2.5)	0.327 (0.17)	4000 (0)
Max. iterations	500	1.06 (1.8)	0.277 (0.16)	18085 (252)

Table 4c: Results of applying algorithm 1 on test problem 7

Stopping rule \ Algorithm	1	2	3	4	5	6	7	8
IMPROVE	5	4	5	4	5	4	4	4
CONVERGE	2	2	3	3	3	3	3	3
MAXEVAL	---	---	---	---	---	---	---	---

Table 5: Settings of the different stopping criteria in phase 2

Algorithm 1	NO RESTART		RESTART	
Test Function	**ERROR**	**nEVAL**	**ERROR**	**nEVAL**
Rosenbrock function	9.92 (86)	107 (10)	1.28 (0.40)	20.5 (10)
Powell singular function	19334 (0.0)	34.6 (4.1)	18287 (0.0)	36.9 (8.7)
Gaussian function	0.96 (1.16)	37.7 (13)	0.24 (0.25)	23.1 (8.5)
Parabolic function	1102 (3524)	346 (136)	120 (1200)	109 (89)
Asymmetric function	7.83 (19.8)	1411 (654)	0.33 (0.36)	853 (605)
Beale function	1207 (701)	38.6 (35)	679 (739)	41.0 (35)
Wood function	4203 (3174)	199 (244)	2664 (3069)	188 (239)
Simulation model	0.109 (0.11)	28.5 (9.1)	0.094 (0.099)	25.1 (5.3)

Table 6a: Results of applying the Fu version of algorithm 1

Algorithm 2	NO RESTART		RESTART	
Test Function	**ERROR**	**nEVAL**	**ERROR**	**nEVAL**
Rosenbrock function	44.7 (190)	103 (19)	1.25 (0.37)	24.4 (19)
Powell singular function	19345 (76)	34.6 (4.1)	18328 (141)	35.7 (6.9)
Gaussian function	1.36 (1.47)	33.8 (12)	0.31 (0.33)	24.6 (9.8)
Parabolic function	1235 (3724)	334 (94)	0.30 (0.23)	113 (90)
Asymmetric function	3.68 (13.8)	1436 (542)	0.37 (0.32)	844 (558)
Beale function	1134 (746)	42.9 (36)	672 (746)	38.6 (34)
Wood function	4026 (3219)	207 (264)	2184 (2965)	230 (251)
Simulation model	0.116 (0.15)	25.9 (5.7)	0.089 (0.107)	25.0 (4.2)

Table 6b: Results of applying the Fu version of algorithm 2

Algorithm 5	NO RESTART		RESTART	
Test Function	**ERROR**	**nEVAL**	**ERROR**	**nEVAL**
Rosenbrock function	69.6 (233)	374 (106)	1.13 (0.21)	50.0 (101)
Powell singular function	19334 (0.0)	37.2 (9.1)	18287 (0.0)	34.9 (5.0)
Gaussian function	1.05 (1.23)	48.0 (19)	0.29 (0.32)	23.6 (13)
Parabolic function	735 (2924)	1187 (282)	120 (1200)	141 (254)
Asymmetric function	4.72 (15.5)	1700 (604)	0.88 (5.7)	917 (589)
Beale function	1160 (727)	117 (159)	620 (732)	120 (155)
Wood function	4274 (3148)	431 (597)	2609 (3053)	416 (548)
Simulation model	0.094 (0.11)	27.5 (9.6)	0.085 (0.108)	25.0 (4.2)

Table 6c: Results of applying the Fu version of algorithm 5

Algorithm 6	NO RESTART		RESTART	
Test Function	**ERROR**	**nEVAL**	**ERROR**	**nEVAL**
Rosenbrock function	70.7 (237)	372 (103)	1.11 (0.18)	49.7 (102)
Powell singular function	19345 (76)	34.6 (4.1)	18315 (127)	35.2 (7.0)
Gaussian function	1.29 (1.42)	46.2 (21.2)	0.31 (0.43)	25.6 (14.1)
Parabolic function	864 (3164)	1182 (307)	0.23 (0.17)	160 (296)
Asymmetric function	2.40 (10.8)	1688 (502)	0.34 (0.29)	861 (675)
Beale function	1164 (730)	117 (158)	685 (746)	109 (150)
Wood function	4867 (2918)	358 (606)	3117 (3135)	439 (572)
Simulation model	0.107 (0.12)	26.5 (7.2)	0.089 (0.081)	26.9 (7.3)

Table 6d: Results of applying the Fu version of algorithm 6

	Algorithms							
Test function	1	2	3	4	5	6	7	8
Rosenbrock function	1.13 (0.21)	1.14 (0.21)	1.16 (0.27)	1.19 (0.49)	1.03 (0.16)	1.07 (0.16)	1.04 (0.10)	1.08 (0.17)
Powell sing. function	0.27 (0.28)	0.86 (0.93)	0.25 (0.24)	1.09 (1.70)	0.24 (0.23)	1.43 (5.90)	0.26 (0.29)	0.77 (0.82)
Gaussian function	0.19 (0.20)	0.18 (0.19)	0.17 (0.20)	0.16 (0.17)	0.16 (0.15)	0.17 (0.16)	0.19 (0.21)	0.19 (0.21)
Parabolic function	0.24 (0.16)	0.23 (0.17)	0.22 (0.17)	0.27 (0.35)	0.20 (0.13)	0.23 (0.17)	0.21 (0.13)	0.22 (0.16)
Asymmetric function	0.18 (0.15)	0.25 (0.17)	0.20 (0.16)	0.55 (2.71)	0.21 (0.18)	0.31 (0.32)	0.18 (0.15)	0.25 (0.19)
Beale function	0.25 (0.18)	0.64 (2.91)	0.31 (0.25)	0.40 (0.63)	0.20 (0.10)	0.23 (0.22)	0.24 (0.14)	0.28 (0.33)
Wood function	1.44 (1.9)	28.1 (37)	4.77 (7.8)	32.4 (36)	0.81 (0.87)	20.7 (28)	3.16 (5.7)	24.5 (30)
Simulation model	0.067 (0.081)	0.11 (0.13)	0.061 (0.074)	0.089 (0.099)	0.068 (0.060)	0.090 (0.11)	0.073 (0.098)	0.094 (0.12)

Table 7a: Mean and standard deviation of the errors of the eight algorithms using the new stopping criteria

	Algorithms							
Test function	1	2	3	4	5	6	7	8
Rosenbrock function	0.849 (0.093)	0.918 (0.028)	0.826 (0.070)	0.893 (0.061)	0.758 (0.157)	0.874 (0.040)	0.875 (0.042)	0.850 (0.083)
Powell sing. function	0.020 (0.012)	0.057 (0.022)	0.019 (0.013)	0.087 (0.032)	0.021 (0.011)	0.050 (0.018)	0.023 (0.012)	0.049 (0.024)
Gaussian function	0.010 (0.006)	0.006 (0.006)	0.003 (0.002)	0.009 (0.006)	0.009 (0.007)	0.016 (0.007)	0.012 (0.007)	0.025 (0.012)
Parabolic function	0.047 (0.020)	0.050 (0.013)	0.037 (0.016)	0.049 (0.014)	0.046 (0.017)	0.053 (0.018)	0.059 (0.017)	0.056 (0.017)
Asymmetric function	0.035 (0.010)	0.054 (0.021)	0.032 (0.012)	0.042 (0.011)	0.040 (0.006)	0.051 (0.016)	0.030 (0.011)	0.059 (0.008)
Beale function	0.127 (0.002)	0.124 (0.002)	0.129 (0.002)	0.127 (0.001)	0.123 (0.005)	0.126 (0.003)	0.125 (0.003)	0.127 (0.001)
Wood function	0.108 (0.042)	0.786 (0.449)	0.083 (0.039)	1.882 (0.968)	0.077 (0.050)	0.363 (0.187)	0.106 (0.052)	0.445 (0.245)
Simulation model	0.006 (0.003)	0.004 (0.003)	0.004 (0.002)	0.003 (0.001)	0.005 (0.003)	0.006 (0.003)	0.005 (0.002)	0.004 (0.002)

Table 7b: Mean and standard deviation of the ten smallest errors of the eight algorithms using the new stopping criteria

	Algorithms							
Test function	1	2	3	4	5	6	7	8
Rosenbrock function	798 (252)	582 (189)	405 (157)	318 (123)	972 (230)	844 (197)	686 (208)	659 (179)
Powell sing. function	4475 (2471)	2612 (846)	4239 (1951)	2391 (820)	5257 (2884)	3915 (1047)	4166 (1734)	3606 (906)
Gaussian function	1382 (777)	1026 (450)	662 (275)	568 (167)	1576 (701)	1131 (521)	685 (288)	629 (241)
Parabolic function	6151 (5080)	3727 (2141)	4968 (4035)	2784 (1559)	7096 (5264)	4516 (2084)	4313 (2393)	4000 (1844)
Asymmetric function	31179 (26362)	18255 (11874)	31121 (21364)	18611 (12144)	32815 (27925)	18844 (9987)	19003 (12915)	16269 (9588)
Beale function	800 (332)	632 (231)	398 (165)	382 (145)	935 (260)	827 (221)	680 (212)	660 (220)
Wood function	5319 (2041)	2904 (1101)	2701 (1116)	2138 (861)	6626 (2780)	3973 (1115)	4022 (1068)	3646 (1239)
Simulation model	817 (420)	596 (273)	202 (113)	204 (118)	501 (250)	402 (224)	184 (113)	207 (136)

Table 7c: Mean and standard deviation of the number of evaluations of the eight algorithms using the new stopping criteria

	Algorithms							
Test function	1	2	3	4	5	6	7	8
Rosenbrock function	1.10 (0.18)	1.09 (0.17)	1.11 (0.21)	1.10 (0.18)	1.02 (0.15)	1.05 (0.15)	1.03 (0.11)	1.05 (0.16)
Powell sing. function	0.20 (0.19)	0.36 (0.32)	0.21 (0.19)	0.35 (0.27)	0.19 (0.18)	0.35 (0.28)	0.22 (0.19)	0.34 (0.28)
Gaussian function	0.15 (0.15)	0.16 (0.17)	0.15 (0.14)	0.13 (0.15)	0.14 (0.14)	0.14 (0.11)	0.15 (0.17)	0.15 (0.16)
Parabolic function	0.22 (0.15)	0.22 (0.17)	0.21 (0.17)	0.22 (0.15)	0.19 (0.12)	0.23 (0.16)	0.21 (0.13)	0.21 (0.15)
Asymmetric function	0.18 (0.15)	0.23 (0.17)	0.19 (0.16)	0.24 (0.18)	0.19 (0.16)	0.21 (0.16)	0.14 (0.11)	0.20 (0.14)
Beale function	0.22 (0.13)	0.23 (0.17)	0.28 (0.24)	0.28 (0.22)	0.20 (0.10)	0.20 (0.11)	0.21 (0.11)	0.25 (0.18)
Wood function	0.47 (0.38)	5.39 (7.26)	1.18 (2.67)	5.85 (6.74)	0.41 (0.43)	3.19 (4.54)	0.85 (1.33)	5.93 (12.9)
Simulation model	0.061 (0.056)	0.090 (0.096)	0.056 (0.055)	0.081 (0.081)	0.066 (0.059)	0.064 (0.059)	0.069 (0.094)	0.075 (0.065)

Table 8a: Mean and standard deviation of the errors of the eight algorithms using the new stopping criteria and the restart mechanism

	Algorithms							
Test function	1	2	3	4	5	6	7	8
Rosenbrock function	0.836 (0.082)	0.829 (0.099)	0.805 (0.070)	0.851 (0.044)	0.731 (0.146)	0.855 (0.031)	0.860 (0.041)	0.776 (0.162)
Powell sing. function	0.019 (0.010)	0.043 (0.012)	0.016 (0.012)	0.037 (0.017)	0.018 (0.009)	0.037 (0.015)	0.022 (0.010)	0.034 (0.017)
Gaussian function	0.008 (0.005)	0.006 (0.005)	0.003 (0.002)	0.006 (0.004)	0.006 (0.004)	0.009 (0.005)	0.009 (0.005)	0.011 (0.010)
Parabolic function	0.044 (0.018)	0.048 (0.012)	0.037 (0.016)	0.049 (0.014)	0.044 (0.015)	0.049 (0.017)	0.053 (0.017)	0.058 (0.019)
Asymmetric function	0.034 (0.006)	0.042 (0.015)	0.030 (0.011)	0.042 (0.011)	0.041 (0.005)	0.042 (0.014)	0.030 (0.010)	0.047 (0.012)
Beale function	0.127 (0.002)	0.124 (0.002)	0.128 (0.002)	0.127 (0.001)	0.122 (0.004)	0.125 (0.003)	0.125 (0.003)	0.127 (0.001)
Wood function	0.076 (0.025)	0.280 (0.137)	0.057 (0.024)	0.314 (0.243)	0.029 (0.019)	0.193 (0.055)	0.057 (0.021)	0.145 (0.063)
Simulation model	0.006 (0.003)	0.004 (0.003)	0.004 (0.002)	0.003 (0.001)	0.004 (0.002)	0.006 (0.002)	0.006 (0.002)	0.005 (0.003)

Table 8b: Mean and standard deviation of the ten smallest errors of the eight algorithms using the new stopping criteria and the restart mechanism

	Algorithms							
Test function	1	2	3	4	5	6	7	8
Rosenbrock function	655 (248)	492 (175)	275 (151)	250 (121)	582 (260)	460 (185)	316 (163)	247 (145)
Powell sing. function	3192 (3122)	1707 (1179)	2688 (1988)	1475 (886)	3170 (2660)	1710 (967)	1786 (1721)	1469 (818)
Gaussian function	1346 (762)	949 (487)	642 (286)	517 (229)	1332 (723)	1044 (547)	648 (275)	551 (216)
Parabolic function	5107 (4615)	2796 (2111)	4446 (4039)	2728 (1612)	4206 (3291)	3055 (1892)	3042 (2373)	2573 (1722)
Asymmetric function	23785 (18956)	13284 (8856)	23214 (19591)	13238 (8217)	27503 (23980)	15834 (9546)	17481 (15337)	14311 (9591)
Beale function	686 (291)	528 (214)	300 (185)	251 (140)	602 (243)	443 (174)	267 (135)	244 (141)
Wood function	3266 (2432)	2148 (862)	1856 (1357)	1584 (903)	3155 (2293)	2052 (1022)	1572 (1075)	1600 (873)
Simulation model	717 (323)	583 (281)	166 (113)	202 (135)	480 (278)	370 (183)	163 (91)	172 (128)

Table 8c: Mean and standard deviation of the number of evaluations done in the restart of the eight algorithms using the new stopping criteria

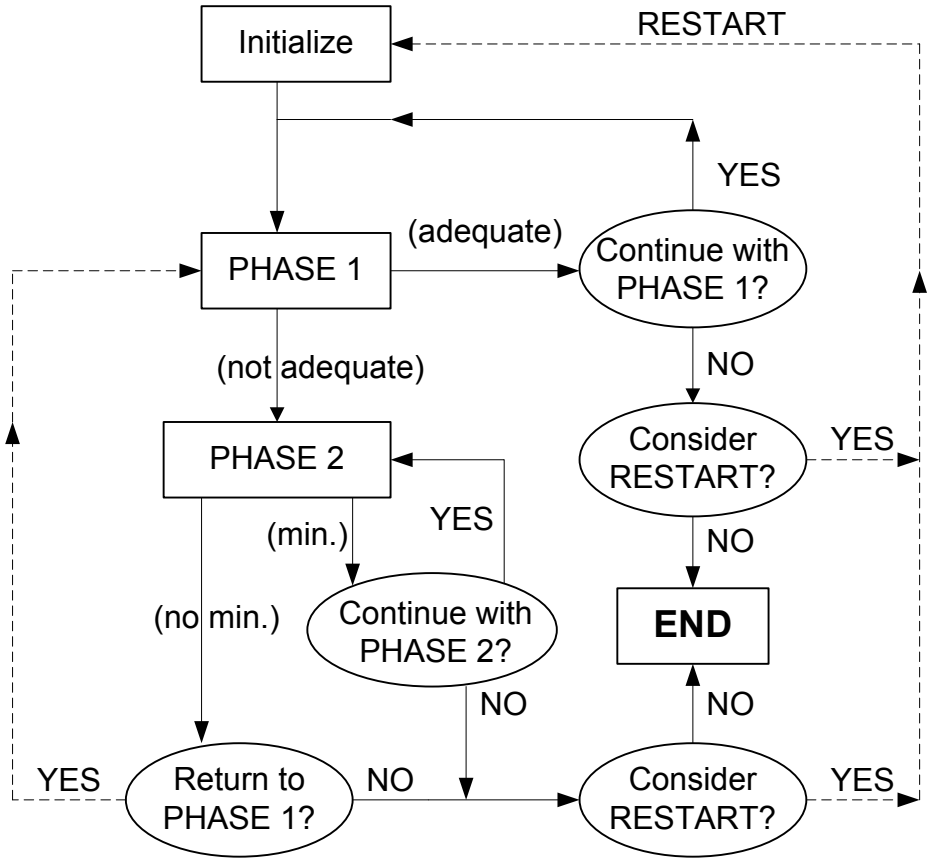


Figure 1: Schematic overview of the algorithm
Note: min. = minimum

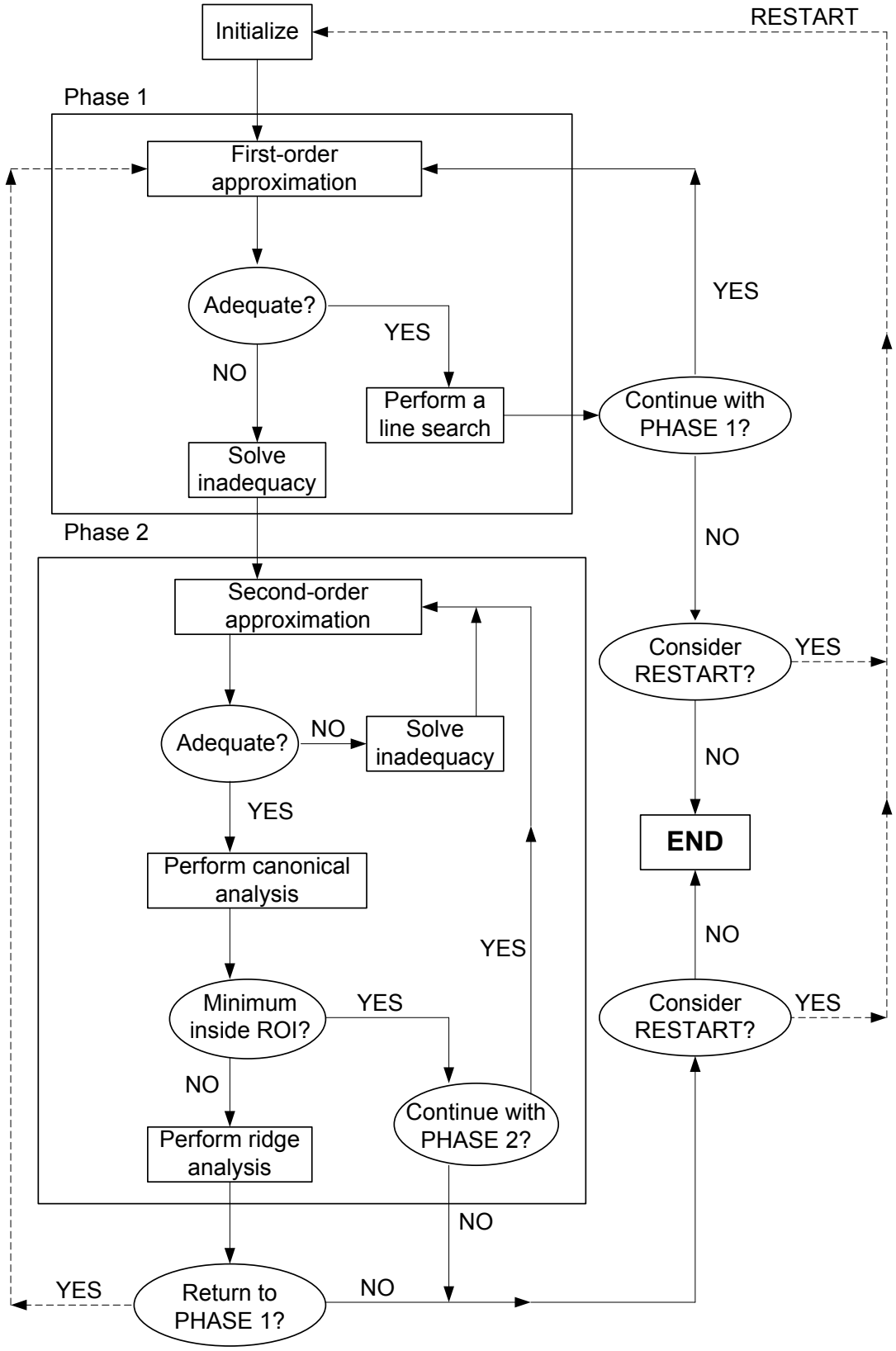
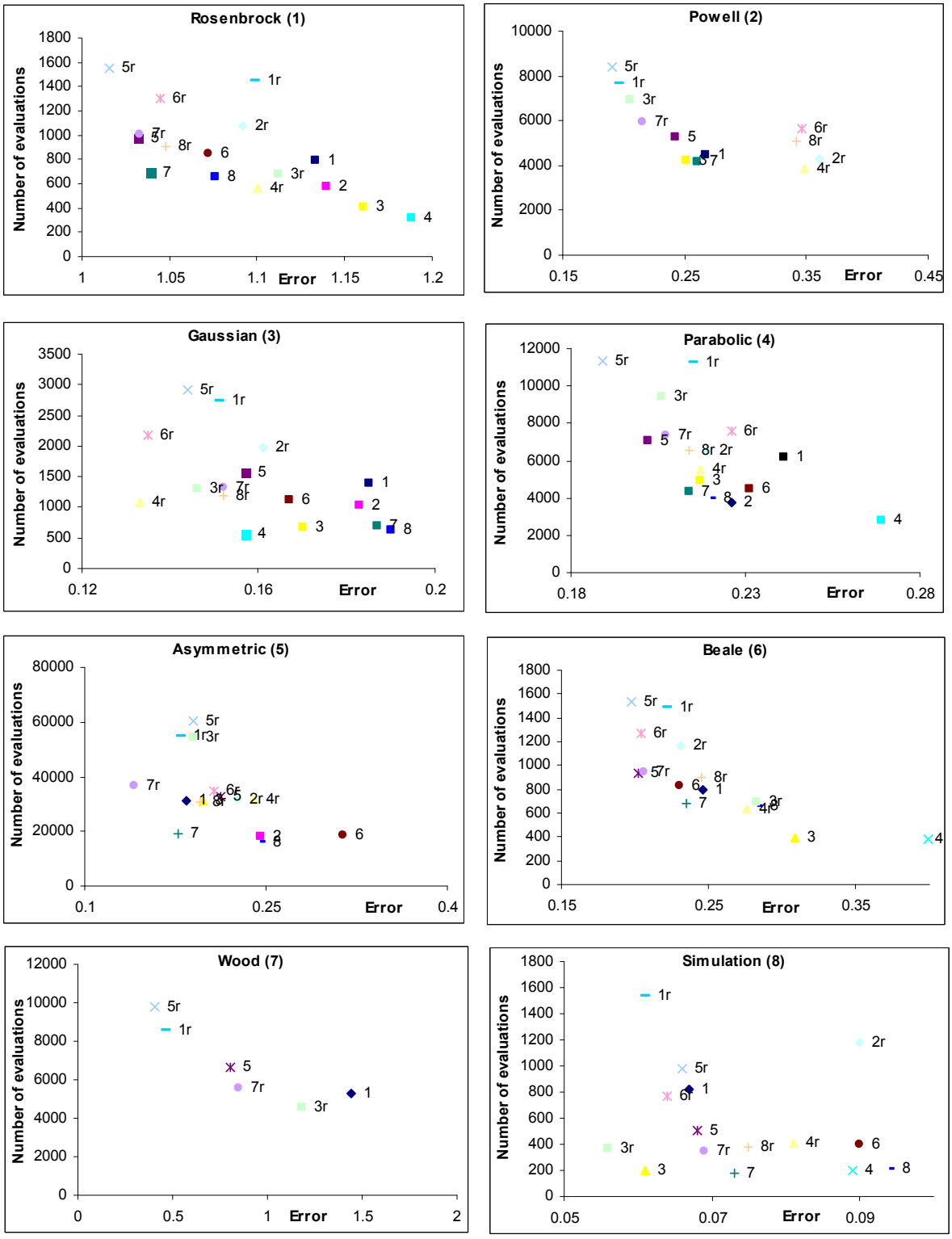
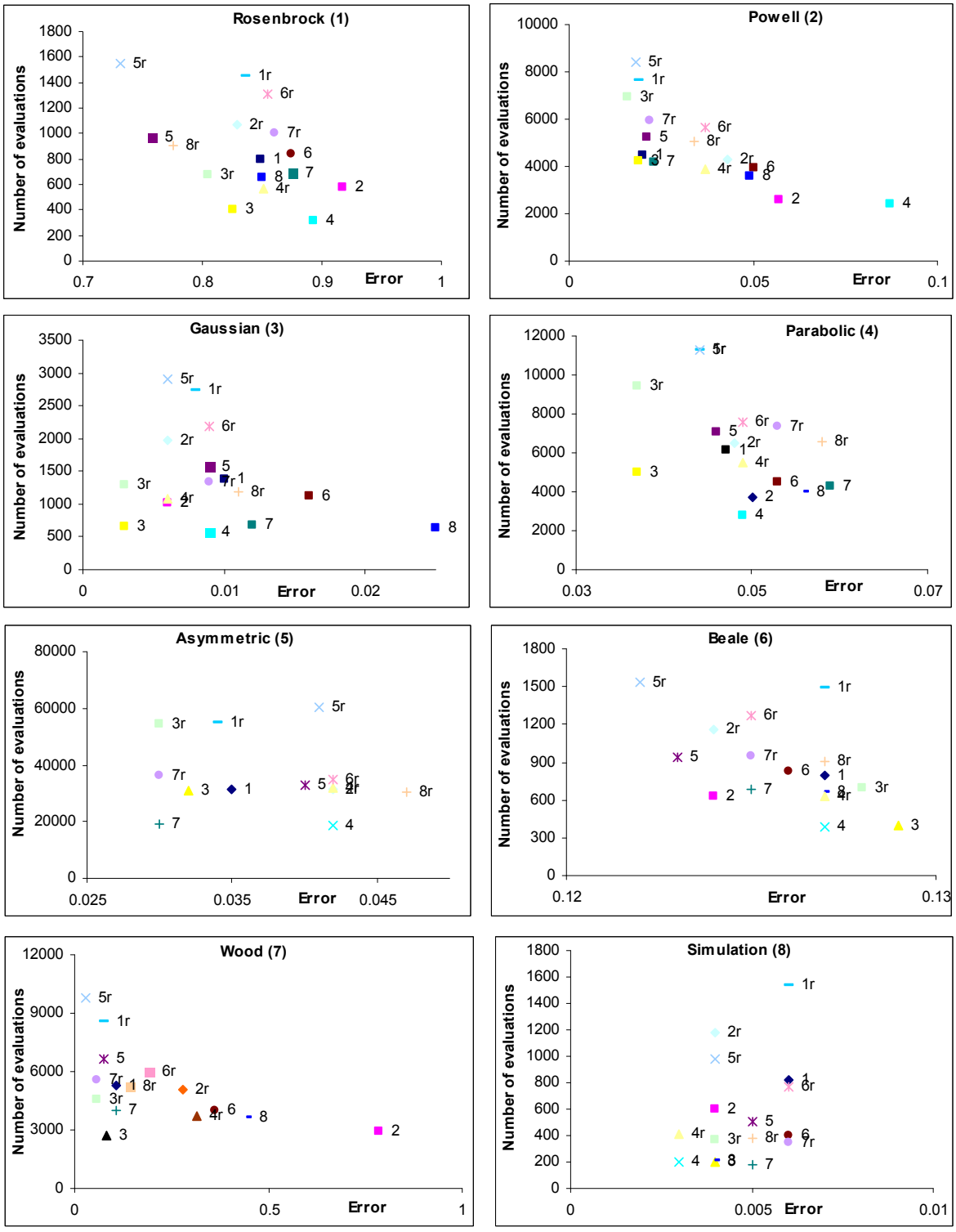


Figure 2: Framework for automated RSM algorithms with building blocks, strategic choices, stopping rules, and a restart mechanism. The building blocks are given by rectangles, the strategic choices are given by ellipses. Note that the decision to continue or to restart the algorithm is also a strategic choice.



Figures 3a-h: Mean number of evaluations versus mean error of algorithms
1r denotes algorithm 1 using the restart mechanism



Figures 4a-h: Mean number of evaluations versus mean error of best 10 solutions
1r denotes algorithm 1 using the restart mechanism