CREATES
Center for Research in Econometric Analysis of Time Series

CREATES Research Paper 2008-32

How to Maximize the Likelihood Function for a DSGE Model

Martin Møller Andreasen

School of Economics and Management
University of Aarhus
Building 1322, DK-8000 Aarhus C
Denmark

Aarhus School of Business
University of Aarhus
Handelshøjskolen
Aarhus Universitet

UNIVERSITY OF
COPENHAGEN

How to Maximize the Likelihood Function for a DSGE Model

Martin Møller Andreasen*
University of Aarhus and CREATES
School of Economics and Management

June 18, 2008

Abstract

This paper extends two optimization routines to deal with objective functions for DSGE models. The optimization routines are i) a version of Simulated Annealing developed by Corana, Marchesi & Ridella (1987), and ii) the evolutionary algorithm CMA-ES developed by Hansen, Müller & Koumoutsakos (2003). Following these extensions, we examine the ability of the two routines to maximize the likelihood function for a sequence of test economies. Our results show that the CMA-ES routine clearly outperforms Simulated Annealing in its ability to find the global optimum and in efficiency. With 10 unknown structural parameters in the likelihood function, the CMA-ES routine finds the global optimum in 95% of our test economies compared to 89% for Simulated Annealing. When the number of unknown structural parameters in the likelihood function increases to 20 and 35, then the CMA-ES routine finds the global optimum in 85% and 71% of our test economies, respectively. The corresponding numbers for Simulated Annealing are 70% and 0%.

Keywords: CMA-ES optimization routine, Multimodel objective function, Nelder-Mead simplex routine, Non-convex search space, Resampling, Simulated Annealing.

JEL: C61, C88, E30

*Email: mandreasen@econ.au.dk. Telephone number: +45 8942 2138. I greatly acknowledge financial support from the Danish Center for Scientific Computation (DCSC). I also appreciate financial support to Center for Research in Econometric Analysis of Time Series, CREATES, funded by the Danish National Research Foundation. A special thanks to Nikolaus Hansen and Alan Miller for providing codes of the existing CMA-ES routine and the Simulated Annealing routine, respectively, on their homepages. Finally, I am also thankful to my advisors Bent Jesper Christensen, Torben M. Andersen, and Henning Bunzel for useful comments. First version of this paper: February 12, 2008.

1 Introduction

During the last 20 years, DSGE models have become one of the leading frameworks for developing micro founded macroeconomic models. From the simple neoclassical growth model, large DSGE models have been developed. Recent examples are the models by Smets & Wouters (2003), Christiano, Eichenbaum & Evans (2005), Altig, Christiano, Eichenbaum & Linde (2005), and Justiniano & Primiceri (2008). The construction and evaluation of these large models have fuelled a desire among researchers to estimate a subset or all of the structural parameters in these models. However, DSGE models are very complex due to the use of rational expectations and the many restrictions imposed on the structural parameters. For instance, a unique and stable solution does not exist for all combinations of the structural parameters in many DSGE models. Thus, restricting the focus to unique and stable solutions may imply that the parameter space for these DSGE models is a non-convex set. A non-convex and high dimensional parameter space severely complicates any attempt to optimize an objective. Moreover, objective functions of DSGE models often have many local optima, and this fact further complicates the optimization problem. Hence, estimating DSGE models by classical inference, i.e. by Maximum Likelihood, GMM, SMM, or Indirect Inference, is a very challenging task. The renewed focus on Bayesian estimation methods have so far been a way to get around this task because Bayesian estimation methods simply update the prior distributions to get the posterior distributions, and this does not involve optimization.

This paper examines the ability of two optimization routines to find the global optimum of the likelihood function for a representative DSGE model at many different parameter configurations. We restrict our focus to global optimization routines that rely on simulation. Two reasons motivate this choice. First, these routines are able to deal with the high dimensional and non-convex parameter space in an easy and intuitive way. Second, these routines do not rely on the presence of first-order or even second-order derivatives of the likelihood function. This feature is convenient if the likelihood function is evaluated by a particle filter, where the resampling step generates discontinuities in the likelihood function (Fernández-Villaverde & Rubio-Ramírez (2007), Rossi (2004)). Thus, a gradient based optimization routine is likely to perform poorly in this case.

We study a version of Simulated Annealing developed by Corana et al. (1987) and the evolutionary algorithm CMA-ES developed by Hansen et al. (2003). The choice of these optimization routines is motivated by the fact that they represent two different principles which are both widely used within the optimization literature. We choose to focus on the version of Simulated Annealing developed by Corana et al. (1987) because Goffe, Ferrier & Rogers (1994) show that it performs well on a model with rational expectations. However, this version of Simulated Annealing is in principle not able to handle a non-convex parameter space so we extend the routine to meet this requirement. This is done by adding a resampling step to the algorithm if an undefined point is located. The CMA-ES routine is chosen because: i) it should be able to handle a non-convex parameter space, ii) it does well on non-separable multimodal objective functions (Hansen & Kern (2004)), and iii) it is better than other evolutionary algorithms on multimodal objective functions (Kern, Muller, Hansen, Buche, Ocenasek & Koumoutsakos (2004)). However, our test study shows that the CMA-ES routine breaks down due to the highly non-convex search space when there are 20 or more unknown structural parameters in the likelihood function. Hence, we extend the CMA-ES algorithm so it can optimize likelihood functions

even for large DSGE models. This is done by modifying the updating rule for the covariance matrix in the evolutionary search distribution.[1]

Our test study reveals a number of interesting results for the extended versions of Simulated Annealing and the extended CMA-ES routine. First, the CMA-ES routine clearly outperforms Simulated Annealing in all tests. That is, the CMA-ES routine finds the global optimum more often than Simulated Annealing does, and in addition the CMA-ES routine uses notably fewer function evaluations than Simulated Annealing. Hence, the CMA-ES routine is the most efficient optimization routine of the two. Second, with 10 unknown structural parameters in the likelihood function, the CMA-ES routine finds the global optimum in 95% our test economies compared to 89% for Simulated Annealing. When the number of unknown structural parameters in the likelihood function increases to 20 and 35, then the CMA-ES routine finds the global optimum in 85% and 71% of the test economies, respectively. The corresponding numbers for Simulated Annealing are 70% and 0%.

We emphasize that this paper does not take a stand in the discussion of whether researchers should use classical or Bayesian estimation methods in relation to DSGE models. The use of optimization algorithms is unavoidable for the classical researcher but also useful for the Bayesian researcher. For instance, the Bayesian estimation procedures described by An & Schorfheide (2007) require a maximization of the likelihood function multiplied by the prior distributions to get good starting values for the MCMC analysis. Hence, the routines presented in this paper should be of interest to both a classical and a Bayesian researcher.

The rest of this paper is organized as follows. Section 2 describes the optimization problem in relation to DSGE models. Section 3 presents the Simulated Annealing algorithm and our extension of this optimization routine. The CMA-ES routine and our modification of this routine are described in section 4. We set up a DSGE model in section 5 and construct a sequence of "test economies". The ability of the optimization routines to maximize the likelihood functions for these test economies is reported in section 6. Section 7 concludes.

2 The optimization problem

We start by describing the problem which must be solved by the two optimization routines. This is done by considering a minimization problem which is without loss of generality. Let $Q(\mathbf{x})$ denote a non-linear objective function mapping from the search space $S \subseteq \mathbb{R}^n$ into $\mathbb{R}$. Then the problem is

$$\underset{\mathbf{x} \in S}{Min}\ Q(\mathbf{x}). \tag{1}$$

In our case, $Q(\mathbf{x}) = -\mathcal{L}(\mathbf{x})$, where $\mathcal{L}(\mathbf{x})$ is the likelihood function, and the n-dimensional vector $\mathbf{x}$ contains the structural parameters of the DSGE model. The search space (S) is constructed based on constraints related to the model. In addition to the constraints ensuring unique and stable solutions, many of the structural parameters are also subject to sign restrictions. Moreover, the non-negativity restrictions on quantities such as consumption, production, etc. may

[1] Fortran and Matlab versions of the two extended optimization routines are available from the author's homepage.

also impose restrictions on the structural parameters. Furthermore, conditions for ensuring finite objective functions for firms and households also reduce the parameter space. Finally, we have to take into account that in general an explicit expression for the search space cannot be derived because DSGE models are solved numerically. We take note of this fact and henceforth only require that we are able to evaluate numerically whether a particular value of $\mathbf{x}$ is in S.

Before we describe the two optimization routines, a small technical comment is in order. One attempt to deal with points outside S might be to simply assign the objective function an arbitrary large value when $\mathbf{x} \notin S$. This ensures that the minimization routine never searches for optimal points outside S and furthermore generates a convex search space. The latter implies that standard optimization routines can be used. However, these optimization routines should not be gradient based routines because the coding generates strong kinks in the objective function. Furthermore, this procedure also has the drawback that it may generate artificial local minima in the objective function and thus make the objective function even harder to optimize. We return to this point in the next sections, and we show that assigning undefined points an arbitrary large value is suboptimal and may reduce the performance of the two optimization routines considered.

3 Simulated Annealing

3.1 The basic algorithm

The idea behind the Simulated Annealing algorithm is to mimic the behavior of a physical system in thermodynamic equilibrium which when cooled sufficiently slowly (annealing) converges to the state with the lowest energy level, i.e. the lowest value of the objective function. More specifically, Simulated Annealing works by letting "a random walker" move around in the n dimensional search space according to the Metropolis criterion. This criterion always accepts moves which decrease the value of the objective function ($\triangle Q < 0$), and uphill moves ($\triangle Q > 0$) are accepted with the probability $\exp\{- \triangle Q/T\}$. Here, T is a parameter which denotes the temperature of the system. Thus, a low temperature value and/or a large increase in the objective function increase the probability of rejecting an uphill move. Because some uphill moves are accepted for $T > 0$, the algorithm is able to escape from local optima. By letting the temperature gradually tend to zero, more and more uphill moves are rejected and the algorithm gradually zooms in on the global minimum at successful termination (Salamon, Sibani & Frost (2002)).

The defining feature for the version of Simulated Annealing developed by Corana et al. (1987) is the sampling from n uniform and independent probability distributions in order to explore the search space. The objective function is required to be bounded and well-defined in the entire search space, but it does not need to be smooth or even continuous. The search space is defined by letting $a(h) \leq x(h) \leq b(h)$ for all h, where $a(h)$ and $b(h)$ are the lower and upper bounds, respectively. In addition to these bounds, a starting value $\mathbf{x} = \begin{bmatrix} x(1) & ... & x(n) \end{bmatrix} \in S$ and initial step sizes $\mathbf{v} = \begin{bmatrix} v(1) & ... & v(n) \end{bmatrix}$ must be provided. The algorithm has five parameters $(N_T, N_S, r_T, T, N_\epsilon)$, and we discuss proper values for each of them after the presentation of the algorithm.

In order to explain how the algorithm works, a pseudo-code for one iteration is presented in figure 1.

< Figure 1 about here >

Let $m = 1$ and $j = 1$ and look at the loop indexed by h which runs through all the n parameters in the objective function. For the first parameter $(h = 1)$, we draw from the uniform distribution $u(h) \in [-1, 1]$ scaled by the step size $v(h)$. The values for all the remaining parameters of the objective function are unchanged. The change in the parameters from $\mathbf{x}$ to $\mathbf{x}'$ is referred to as a "move" for the random walker. If $x'(h) \notin [a(h), b(h)]$, a new value for $x'(h)$ is generated by drawing uniformly from $[a(h), b(h)]$. Based on the Metropolis criterion, the suggested move to $\mathbf{x}'$ is either accepted or rejected. The procedure is then repeated for the second parameter $(h = 2)$, the third parameter $(h = 3)$, and so forth. Thus, when the loop indexed by h is completed, attempts have been made to move along each of the n directions in the search space. We repeat this procedure N_S times, meaning that we let the random walker travel N_S times through every direction of the search space. After this "walk" for the first random walker, the variances for each of the n search distributions are adjusted such that approximately 50% of the suggested moves are accepted. This is done by decreasing the variance for the j'th distribution if too many suggested moves are rejected along the j'th dimension of the search space, and vice versa. Corana et al. (1987) implement this criterion by the rule

$$p(h) = \frac{\text{Number of accepted moves in direction } h}{\text{Number of accepted and rejected moves in direction } h} \tag{2}$$

$$\tilde{v}(h)_{new} = \begin{cases} v(h)\left(1 + 2\frac{p(h)-0.6}{0.4}\right) & \text{if } p(h) > 0.6 \\ v(h)\left(\frac{1}{1+2\frac{0.4-p(h)}{0.6}}\right) & \text{if } p(h) < 0.4 \\ v(h) & \text{otherwise} \end{cases} \quad \text{for all } h \tag{3}$$

$$v(h)_{new} = \begin{cases} \tilde{v}(h)_{new} & \tilde{v}(h)_{new} \leq b(h) - a(h) \\ b(h) - a(h) & \text{otherwise} \end{cases} \quad \text{for all } h \tag{4}$$

After having adjusted the n search distributions, the second random walker $(m = 2)$ takes a similar walk in the search space. The first iteration is complete when all N_T random walkers have completed their walks in the search space. We denote the value of the objective function after all N_T random walkers by Q^* and the optimal value of the objective function so far by Q^*_{opt}.

Before the start of the next iteration, the parameter controlling the temperature is reduced to $T_{new} = r_T T$ where $r_T \in (0, 1)$. With this new temperature, the procedure above is repeated and started at $\mathbf{x} = \mathbf{x}^*_{opt}$ where $Q^*_{opt} = Q\left(\mathbf{x}^*_{opt}\right)$. The lower temperature makes uphill move less likely and thereby increases the number of rejected moves, leading to smaller step sizes. The algorithm is terminated when $Q^* - Q^*_{opt} \leq \epsilon$ and $|Q^* - Q^*_u| \leq \epsilon_Q$ for $u = 1, ..., N_\epsilon$ where ϵ is a small number, and Q^*_u denotes the value of the objective function from the u'th previous iteration.[2]

[2] We refer to Corana et al. (1987) for a full description of the algorithm for Simulated Annealing in pseudo-codes.

Thus, the basic idea behind Simulated Annealing is first to explore the entire search space. This is done with large step sizes and a high temperature, which implies that many uphill moves are accepted. As the temperature and the step sizes are gradually decreased, the routine zooms in on the most promising area of the search space. The two parameters N_S and N_T determine how well the objective function is explored for a given temperature and how often the variances in the search distributions are adjusted. Corana et al. (1987) recommend $N_S = 20$ and $N_T = \max(100, 5n)$. The value for N_T is rather large and motivated by the fact that the theoretical foundation for Simulated Annealing requires simulating the frequency distribution for the objective function (the so-called Boltzmann distribution) for each temperature of the system. A too low value of N_T may give a bad approximation of this distribution and hence reduce the performance of the algorithm.

The parameter r_T determines how slowly the temperature is reduced. Therefore, this parameter is also of great importance because the theoretical foundation for Simulated Annealing requires that the system is cooled sufficiently slowly in order to ensure that the routine terminates in the global minimum. If the system is cooled too quickly, Simulated Annealing might terminate in a local minimum. Corana et al. (1987) recommend $r_T = 0.85$.

The initial temperature (T) must be set such that the objective function is well searched in the beginning. Goffe et al. (1994) suggest setting the initial temperature such that the step sizes have the desired length after the first 2-4 iterations. If the step sizes after these iterations are too small, then the initial temperature should be increased and vice versa. Finally, Corana et al. (1987) recommend letting $N_\epsilon = 4$ and $\epsilon_Q = 10^{-4}$ in the stopping criteria.

3.2 Simulated Annealing for DSGE models

The present version of Simulated Annealing is, strictly speaking, not able to handle points in the search space where the objective function is undefined. This could be points in the search space where the DSGE model does not have a stable and unique solution, or where other constraints are violated. We extend the routine to apply to non-convex search spaces simply by resampling undefined points. That is, if a point is not in the search space, then we discharge the point and draw a new point. In this way, we say that the random walker is replaced by the "informed" random walker because the latter is informed about the domain of the DSGE model. To control the time spent on resampling, an upper bound (N_r) is imposed on the number of times we are willing to resample along a given dimension. If this upper bound is reached, then we do not move along the h'th dimension. Instead, we try to find a move along dimension $h + 1$, or if $h = n$, a new "walk" is started.

The alternative to resampling undefined points is to give the objective function an arbitrary large value at such points and use the version of Simulated Annealing with the "uninformed" random walker. Actually, this is the way the rational expectation model in Goffe et al. (1994) is optimized. However, this procedure is clearly inferior to resampling undefined moves because without resampling, undefined moves are incorrectly recorded as rejected moves even though the moves are undefined. This leads to positive biases in the number of rejected moves which through (3) induces negative biases in the variances for the search distributions. That is, the step sizes may rapidly become too small if the number of undefined moves is large. This may reduce the performance of the routine because too small step sizes result in an improper search of the parameter space, hence increasing the probability of ending up in a local minimum. Note

also that we cannot solve this problem by starting with a very high temperature because of the arbitrary large function value assigned to undefined points in the search space.

4 The CMA-ES optimization routine

4.1 The basic routine

This section presents the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) optimization routine. The basic idea behind the routine is to approximate the inverse of the objective function's Hessian matrix. This is not done based on the use of gradients or second order derivatives but by simulation. Hence, the routine can handle very complicated objective functions which have: i) discontinuities, ii) complex constraints, iii) noise, iv) local optima, and v) a non-convex search space.

The algorithm works in the following way. For each iteration, a set of λ points in the search space is found by sampling from an n-dimensional multivariate normal distribution. These λ points constitute a generation. Thus, generation $g+1$ is created by

$$\mathbf{x}_i^{(g+1)} \sim \mathcal{N}\left(\langle \mathbf{x} \rangle_w^{(g)}, {\sigma^{(g)}}^2 \mathbf{C}^{(g)}\right) \qquad i = 1, ..., \lambda, \tag{5}$$

where $\langle \mathbf{x} \rangle_w^{(g)} \in \mathbb{R}^n$ and ${\sigma^{(g)}}^2 \mathbf{C}^{(g)} \in \mathbb{R}^{n \times n}$ are the mean vector and the covariance matrix, respectively, in generation g. The lower and upper bounds for the search space are denoted by $a(h) \leq x(h) \leq b(h)$ for all h. If $\mathbf{x}_i^{(g+1)}$ violates these bounds or any of the other constraints, then the routine resamples this point until a value of $\mathbf{x}_i^{(g+1)}$ is found where the objective function is defined. Based on the λ points in generation $g+1$, the mean vector and the covariance matrix are updated to improve the search in the next iteration where generation $g+2$ is created in a similar manner.

The mean is updated based on the best $\mu \leq \lambda$ points from generation $g+1$ according to

$$\langle \mathbf{x} \rangle_w^{(g+1)} = \textstyle\sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}^{(g+1)}, \tag{6}$$

where the weights $\{w_i\}_{i=1}^{\mu}$ are strictly positive and sum to one. The notation $\mathbf{x}_{i:\lambda}^{(g+1)}$ refers to the point which yields the i'th lowest value of the objective function among the λ points in generation $g+1$.

The objective when updating the covariance matrix is to fit the search distribution to the contour lines of the objective function. However, it is a difficult task to estimate the $(n+1)\,n/2$ parameters in the covariance matrix in a fast and reliable way. The procedure adopted in the CMA-ES routine is to split the problem into two parts and update the unscaled covariance matrix $\left(\mathbf{C}^{(g)}\right)$ and the global step size $\left(\sigma^{(g)}\right)$ separately. Convenient starting values for each of these parameters is to let $\sigma^{(0)} = 1$ and let $\mathbf{C}^{(0)}$ be a diagonal matrix with the appropriate variances. That is, the search is started by sampling from n independent normal distributions where the variances are determined such that we get a proper search along each of the n dimensions.

The update of $\mathbf{C}^{(g)}$ is given by

$$\mathbf{C}^{(g+1)} = \underbrace{(1-c_{cov})\,\mathbf{C}^{(g)}}_{\text{term 1}} + \frac{c_{cov}}{\mu_{cov}} \underbrace{\left(\mathbf{p}_c^{(g+1)}\left(\mathbf{p}_c^{(g+1)}\right)' + \left(1-H_\sigma^{(g+1)}\right)c_c\,(2-c_c)\right)}_{\text{term 2}} \tag{7}$$

$$+c_{cov}\underbrace{\left(1-\frac{1}{\mu_{cov}}\right)\sum_{i=1}^{\mu}\frac{w_i}{\sigma^{(g)^2}}\left(\mathbf{x}_{i:\lambda}^{(g+1)}-\langle\mathbf{x}\rangle_w^{(g)}\right)\left(\mathbf{x}_{i:\lambda}^{(g+1)}-\langle\mathbf{x}\rangle_w^{(g)}\right)'}_{\text{term 3}}$$

where

$$\mathbf{p}_c^{(g+1)} = (1-c_c)\,\mathbf{p}_c^{(g)} + H_\sigma^{(g+1)}\sqrt{c_c\,(2-c_c)\,\mu_{eff}}\left(\frac{\langle\mathbf{x}\rangle_w^{(g+1)}-\langle\mathbf{x}\rangle_w^{(g)}}{\sigma^{(g)}}\right) \text{ and } \mathbf{p}_c^{(0)} = \mathbf{0}\in\mathbb{R}^n \tag{8}$$

$$H_\sigma^{(g+1)} = \begin{cases} 1 & \text{if } \frac{\left\|\mathbf{p}_\sigma^{(g+1)}\right\|}{\sqrt{1-(1-c_\sigma)^{2(g+1)}}} < \left(1.5+\frac{1}{n-0.5}\right)E\left(\|N(\mathbf{0},\mathbf{I})\|\right) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The point of departure for $\mathbf{C}^{(g+1)}$ is the unscaled covariance matrix in the previous generation, $\mathbf{C}^{(g)}$. This is term 1 in (7). Term 2 in (7) exploits the information from the correlation between generations. This is done by setting up the evolutionary path ($\mathbf{p}_c^{(g+1)}$), which is a sum of successive steps. The term $\mathbf{p}_c^{(g+1)}\left(\mathbf{p}_c^{(g+1)}\right)'$ is of dimension $n\times n$ and has rank 1, and this part of the update is therefore based on rank 1 information. The third term in (7) helps to improve the update of the covariance matrix if the number of points in a generation is large. The third term has rank $\min(n,\mu)$, and this part of the update is thus based on information of higher rank than 1. For further details and interpretations, we refer to Hansen et al. (2003), Muller, Hansen & Koumoutsakos (2002), and Hansen (2005).

The update of the global step size is done based on another evolutionary path, denoted $\mathbf{p}_\sigma^{(g)}$. If the length of this path is short, then consecutive steps of $\mathbf{p}_\sigma^{(g)}$ for $g=\{1,2,...\}$ tend to cancel each other out. That is, consecutive steps of $\mathbf{p}_\sigma^{(g)}$ are negatively correlated, and the step size should be decreased in this case. On the other hand, a long evolutionary path means that consecutive steps of $\mathbf{p}_\sigma^{(g)}$ for $g=\{1,2,...\}$ tend to point in the same direction, producing positively correlated steps. The step size should be increased in this case. The optimal situation is to have consecutive steps of $\mathbf{p}_\sigma^{(g)}$ which are approximately uncorrelated. The length of $\mathbf{p}_\sigma^{(g)}$ is measured in relation to the length of an evolutionary path under a random selection of points.[3] This benchmark is useful because under random selection, consecutive steps of $\mathbf{p}_\sigma^{(g)}$ are uncorrelated as desired. The specific expression for $\mathbf{p}_\sigma^{(g+1)}$ used in the CMA-ES routine is given by

$$\mathbf{p}_\sigma^{(g+1)} = (1-c_\sigma)\,\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma\,(2-c_\sigma)}\mathbf{B}^{(g)}\mathbf{D}^{(g)^{-1}}\left(\mathbf{B}^{(g)}\right)'\frac{\sqrt{\mu_{eff}}}{\sigma^{(g)}}\left(\langle\mathbf{x}\rangle_w^{(g+1)}-\langle\mathbf{x}\rangle_w^{(g)}\right) \tag{10}$$

[3] A random selection of points means that we simply include points of $\mathbf{x}_i^{(g)}$ without paying attention to the function value these points generate. That is, $\mathbf{x}_{i:\lambda}^{(g)} = \mathbf{x}_i^{(g)}$.

where $\mathbf{p}_\sigma^{(0)} = \mathbf{0} \in \mathbb{R}^n$. The orthogonal matrix $\mathbf{B}^{(g)}$ and the diagonal matrix $\mathbf{D}^{(g)}$ are derived from a principal component analysis of $\mathbf{C}^{(g)}$, i.e. $\mathbf{C}^{(g)} = \mathbf{B}^{(g)}\mathbf{D}^{(g)^2}\left(\mathbf{B}^{(g)}\right)'$. For any realized sequence of $\left\{\mathbf{C}^{(g)}\right\}_{g=1}^{\infty}$, it holds that $\mathbf{p}_\sigma^{(g+1)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ under random selection, given $\mathbf{p}_\sigma^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Hence, for the update of $\sigma^{(g)}$, the length of $\mathbf{p}_\sigma^{(g+1)}$ is compared to its expected length $E(\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|)$ by the relation

$$\ln\left(\sigma^{(g+1)}\right) = \ln\left(\sigma^{(g)}\right) + \frac{c_\sigma}{d_\sigma E(\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|)}\left(\left\|\mathbf{p}_\sigma^{(g+1)}\right\| - E(\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|)\right) \tag{11}$$

Thus, if the selection of points in the CMA-ES routine based on their function values induces a too long or too short evolutionary path, then the value of $\sigma^{(g+1)}$ is either increased or decreased, respectively.

The CMA-ES routine terminates successfully if the change in the function value is less than ϵ_Q. The change in the function value is measured by the difference between the largest and the smallest value of the objective function in a sample consisting of the previous generation and the smallest values from the previous $10 + \lceil 30n/\lambda \rceil$ generations. The routine may also terminate successfully if the changes in all of the objective function's parameters are less than $\epsilon_{\mathbf{x}}$. This change is measured by $\sigma \max\left(|\mathbf{p}_c(h)|, \sqrt{C(h,h)}\right)$ for $h = \{1, ..., n\}$.

Default parameter values for the CMA-ES routine are[4]

$$\lambda = 4 + \lfloor 3\ln(n) \rfloor \quad \mu = \lfloor \lambda/2 \rfloor \quad w_{i=1,...,\mu} = \frac{\ln(\mu+1) - \ln(i)}{\sum_{j=1}^{n} \ln(\mu+1) - \ln(j)} \tag{12}$$

$$\mu_{eff} = \frac{1}{\sum_{i=1}^{n} w_i^2} \quad \mu_{cov} = \mu_{eff} \quad c_\sigma = \frac{\mu_{eff} + 2}{n + \mu_{eff} + 3} \tag{13}$$

$$d_\sigma = 1 + 2\max\left(0, \sqrt{\frac{\mu_{eff} - 1}{n+1}} - 1\right) \max\left(0.3, 1 - \frac{n}{\min\left(g^{\max}, \frac{Q_{\text{eval}}^{\max}}{\lambda}\right)}\right) + c_\sigma \tag{14}$$

$$c_c = \frac{4}{4+n} \quad c_{cov} = \frac{1}{\mu_{cov}} \frac{2}{\left(n + \sqrt{2}\right)^2} + \left(1 - \frac{1}{\mu_{cov}}\right) \min\left(1, \frac{2\mu_{eff} - 1}{(n+2)^2 + \mu_{eff}}\right) \tag{15}$$

Here, $g^{\max}$ denotes the maximum number of generations, and $Q_{\text{eval}}^{\max}$ denotes the maximum number of function evaluations. Only the parameter λ should be controlled by the researcher. Hansen & Kern (2004) show that increasing the value of λ improves the global search properties and the robustness of the routine.

No convergence for the routine occurs if: i) the principal component analysis $\mathbf{C}^{(g)} = \mathbf{B}^{(g)}\mathbf{D}^{(g)^2}\left(\mathbf{B}^{(g)}\right)'$ fails, ii) any eigenvalue in $\mathbf{D}^{(g)}$ is negative or extremely large, iii) the maximal number of function evaluations or iterations has been reached, iv) changes in the parameters are larger than a prespecified upper bound, or v) the variances in the search distribution get too small along a given dimension.

[4]These values are taken from version 2.34 of the CMA-ES routine as implemented by Nikolaus Hansen in Matlab.

Finally, we return to the procedure where undefined points are not resampled but assigned an arbitrary large value. Adopting this procedure in the CMA-ES routine corresponds to reducing the effective value of λ because we get fewer real function evaluations in each generation. The lower effective value of λ reduces the global search capabilities of the routine and thereby increases the probability of ending up in a local minimum. The situation is even worse if some of the undefined points enter into the μ best points in a generation which may easily occur at the beginning of a search. The problem is then that the undefined points affect the updated values for the mean and the covariance in the search distribution even though the objective function is not defined for these points. Thus, undefined points may distort the search distribution in an unpredictable way. Hence, it is clearly optimal to resample undefined points in the CMA-ES routine.

4.2 The CMA-ES routine for DSGE models

The updating scheme for the covariance matrix in (7) - (11) performs poorly when optimizing the likelihood function for a DSGE model with 20 or more unknown parameters. This is due to the fact that σ increases very rapidly during the first 5-10 iterations of an optimization process, in particular if λ has a medium or large value. This results in very large values for σ at the beginning of the search. Although a large global step size improves the global search capabilities of the routine, too large step sizes cause problems in our context because they make it extremely time consuming, if at all possible, to avoid undefined points for the objective function. During preliminary tests, it often happened that the optimization routine simply got stuck and never proceeded to the next generation because a high value of σ made it impossible for the routine to avoid undefined points.

Based on these results, we extend the updating scheme for σ in (10) and (11) as follows. Let τ_1 denote the number of times we observe points where $\mathbf{x}_i^{(g)} \notin [\mathbf{a}, \mathbf{b}]$ in generation g, and let τ_2 denote the number of times where other restrictions are violated in generation g. Then we introduce the following adjustment for the value of the global step size (σ):[5]

$$\text{if } \tau_1 > \lambda \times 500 \text{ then } \sigma = 0.9\sigma \text{ and set } \tau_1 = 0 \tag{16}$$

$$\text{if } \tau_2 > \lambda \times 500 \text{ then } \sigma = 0.9\sigma \text{ and set } \tau_2 = 0 \tag{17}$$

In other words, each time we observe more than $\lambda \times 500$ points outside $[\mathbf{a}, \mathbf{b}]$ or more than $\lambda \times 500$ points where other constraints are violated, then the global step size σ is reduced with 10%. The reason we choose to consider points outside $[\mathbf{a}, \mathbf{b}]$ on their own is that these constraints can be examined in a very fast manner without calculating the value of the objective function. The effect of this modification is typically only present in the first 5-10 iterations where the covariance matrix $\mathbf{C}^{(g)}$ is estimated poorly. As an additional precaution against too large values for σ we also impose an upper bound of 10 on σ.[6]

[5] The limit of $\lambda \times 500$ could be increased to $\lambda \times 5000$ or an even higher number at the cost of more function evaluations.

[6] Limiting the upper bound of σ to be 0.1 or an even smaller number may also be useful if a truly local search is needed, for instance in the neighborhood of a potential global optimum.

To evaluate the impact of this change in the updating scheme for σ, a small test is carried out.[7] Here, we use the fact that the original CMA-ES routine works without any problems when there are only 10 unknown parameters in the likelihood function. First, we test the ability of the original version of the CMA-ES algorithm to optimize a set of likelihood functions. Then a similar test is done for the modified version of the CMA-ES routine. Comparing the results, we conclude that the performance of the two versions of CMA-ES routine are almost identical. Hence, the modification of σ in (16) and (17) serves as a convenient adjustment to the update of σ in (10) and (11).

5 A DSGE model

In order to test the optimization routines, a sequence of "test economies" is needed. We construct these economies by considering a DSGE model at many different combinations of the structural parameters. The idea is then to simulate data series from these economies and, based on these data, to optimize the likelihood function. For each of the likelihood functions, we know by construction the true optimum since we know which set of structural parameters generated the simulated data. Thus, the ability of the optimization routines to maximize the likelihood function is then easy to examine. The construction of these test economies is the topic of the next two subsections.

5.1 The model

The DSGE model we use has the same basic structure as the models developed by Smets & Wouters (2003), Christiano et al. (2005), Altig et al. (2005), and Schmitt-Grohé & Uribe (2006). We refer to these papers for additional details. When presenting our model, we use the notation from the macroeconomic literature. Hence, the notation in this section is unrelated to the notation used for the optimization routines in the previous sections.

The households: We assume the existence of a representative family with a continuum of members. The family's preferences are specified by a utility function defined over per capita consumption (c_t) and per capita labor effort (h_t)

$$U_t = E_t \sum_{l=0}^{\infty} \beta^l \varepsilon_{h,t+l} \frac{\left[\left(c_{t+l} - b c_{t-1+l} \right)^{1-\phi_4} \left(1 - h_{t+l} \right)^{\phi_4} \right]^{1-\phi_3} - 1}{1-\phi_3}, \tag{18}$$

where $\beta \in [0,1[$ is the subjective discount factor, $\phi_3 \in]0,1[\cup]1,\infty[$, and $\phi_4 \in]0,1[$. The process for $\varepsilon_{h,t}$ is specified as an AR(1)-process for the gross growth rate $\mu_{\varepsilon_h,t+1} \equiv \varepsilon_{h,t+1}/\varepsilon_{h,t}$, i.e.

$$\ln\left(\mu_{\varepsilon_h,t+1}\right) = \rho_{\varepsilon_h} \ln\left(\mu_{\varepsilon_h,t}\right) + \epsilon_{\varepsilon_h,t+1}, \tag{19}$$

where $\rho_{\varepsilon_h} \in]-1,1[$. The error term $\epsilon_{t+1}^{\varepsilon_h}$ is assumed to be independently and identically distributed according to the normal distribution, denoted $\epsilon_{\varepsilon_h,t+1} \sim \mathcal{NID}\left(0, Var\left(\epsilon_{\varepsilon_h,t+1}\right)\right)$. The parameter $b \in [0,1]$ in (18) specifies the level of the internal habit effect for the consumption

[7] This test is done along the same lines as the tests described in section 6.

good. This good is constructed from a continuum of differentiated goods ($c_{i,t}, i \in [0,1]$) and the aggregation function

$$c_t = \left[\int_0^1 c_{i,t}^{\frac{\eta-1}{\eta}} di \right]^{\frac{\eta}{\eta-1}}. \tag{20}$$

The first constraint on the households is given by the labor supply $\left(h_t^j\right)$ to market $j \in [0,1]$

$$h_t^j = \left(\frac{w_{j,t}}{w_t} \right)^{-\tilde{\eta}} h_t^d, \tag{21}$$

where i) $w_{j,t}$ is the real wage in the j'th labor market, ii) w_t is the real wage index, and iii) h_t^d is a measure of the aggregated labor demand.

The second constraint says that wages are not set optimally in a fraction $\tilde{\alpha} \in [0,1[$ of randomly chosen labor markets. In these labor markets, wages are set according to the rule $W_{j,t} = W_{j,t-1} \left(\mu_{z^*} \pi_{t-1}\right)^{\tilde{\chi}}$. The parameter $\tilde{\chi} \in [0,1]$ measures the degree of indexation to the steady state gross growth rate in real wages (μ_{z^*}) and the gross inflation rate in the previous period ($\pi_{t-1} \equiv P_{t-1}/P_{t-2}$).

The third constraint is the law of motion for the physical capital stock (k_t)

$$k_{t+1} = (1-\delta) k_t + i_t \left(1 - S\left(\frac{i_t}{i_{t-1}} \right) \right). \tag{22}$$

The parameter $\delta \in [0,1]$ is the depreciation rate for the capital stock, and i_t is gross investments. The function $S\left(\frac{i_t}{i_{t-1}}\right) = \frac{\kappa}{2}(\frac{i_t}{i_{t-1}} - \mu_i)^2$ with $\kappa \geq 0$ adds investment adjustment costs to the economy based on changes in the growth rate of investments. The value of μ_i is determined in such a way that there are no adjustment costs along the economy's balanced growth path.

The fourth constraint is the households' real period by period budget constraint

$$\begin{aligned} &E_t r_{t,t+1} x_{t+1}^h + c_t \left(1 + l\left(v_t\right)\right) + \Upsilon_t^{-1} \left(i_t + a\left(u_t\right) k_t\right) + m_t^h + n_t \\ &= \frac{x_t^h + m_{t-1}^h}{\pi_t} + r_t^k u_t k_t + \textstyle\int_0^1 w_{j,t} h_{j,t} dj + \phi_t. \end{aligned} \tag{23}$$

The function $l\left(v_t\right)$ determines the transactional costs imposed on the households based on the velocity $v_t \equiv c_t/m_t^h$, where m_t^h denotes the amount of real money held by the households. Equation (23) also introduces capital adjustment costs through the function $a\left(u_t\right)$, where u_t is the capacity utilization rate of the capital stock. We let

$$l\left(v_t\right) = \phi_1 v_t + \phi_2 / v_t - 2\left(\phi_1 \phi_2\right)^{0.5} \tag{24}$$

$$a\left(u_t\right) = \gamma_1 \left(u_t - 1\right) + \frac{\gamma_2}{2} \left(u_t - 1\right)^2, \tag{25}$$

where $\phi_1 \geq 0$ and ϕ_2 are subject to the constraint that $l\left(v_t\right) \geq 0$, and u_t is normalized to 1 in the steady state. Furthermore, we require that $\gamma_1 \geq 0$ and $\gamma_2 \geq 0$. The left hand side of (23) is the households' total expenditures in period t which are spent on: i) state-contingent claims $\left(E_t r_{t,t+1} x_{t+1}^h\right)$, ii) consumption including transaction costs ($c_t\left[1 + l\left(v_t\right)\right]$), iii) investments and costs of providing capital services to the firms $\left(\Upsilon_t^{-1}\left(i_t + a\left(u_t\right) k_t\right)\right)$, iv) real money holdings

$\left(m_t^h\right)$, and transfers (n_t) to the government. The right hand side of (23) is the households' total wealth in period t which consists of: i) pay-off from state-contingent assets purchased in period $t-1$ $\left(x_t^h/\pi_t\right)$, ii) real money holdings from the previous period $\left(m_{t-1}^h/\pi_t\right)$, iii) income from selling capital services to the firms $\left(r_t^k u_t k_t\right)$, iv) labor income $\left(\int_0^1 w_{j,t}h_{j,t}dj\right)$, and v) dividends received from the firms (ϕ_t).

The firms: The production in the economy is undertaken by a continuum of firms, indexed by $i \in [0,1]$. All firms have access to the same technology given by

$$y_{i,t}^s = \begin{cases} k_{i,t}^\theta \left(z_t h_{i,t}\right)^{1-\theta} - \psi z_t^* & \text{if } k_{i,t}^\theta \left(z_t h_{i,t}\right)^{1-\theta} - \psi z_t^* > 0 \\ 0 & \text{else} \end{cases} \tag{26}$$

where $\theta \in]0,1[$. Here, $k_{i,t}$ and $h_{i,t}$ denotes physical capital and labor service used by the i'th firm, respectively. We assume that the firms earn zero profit in the steady state. The demand at firm i from the j'th labor market $\left(h_{i,t}^j\right)$ is the solution to the problem

$$\underset{h_{i,t}^j \geq 0}{Min}\ Cost = \int_0^1 W_{j,t}h_{i,t}^j dj \quad St. \quad \left[\int_0^1 \left(h_{i,t}^j\right)^{\frac{\tilde{\eta}-1}{\tilde{\eta}}} dj\right]^{\frac{\tilde{\eta}}{\tilde{\eta}-1}} \geq h_{i,t}. \tag{27}$$

Here, $W_{j,t}$ is the nominal wage paid to labor services in labor market j.

The variable z_t in (26) denotes an aggregate neutral technology shock, and z_t^* is a shock to the firms fixed costs, $\psi > 0$. The relation between these shocks is assumed to be $z_t^* \equiv \Upsilon_t^{\theta/(1-\theta)} z_t$. Letting $\mu_{z,t} \equiv z_t/z_{t-1}$ and $\mu_{\Upsilon,t} \equiv \Upsilon_t/\Upsilon_{t-1}$, we assume that

$$\ln\left(\frac{\mu_{z,t+1}}{\mu_z}\right) = \rho_z \ln\left(\frac{\mu_{z,t}}{\mu_z}\right) + \epsilon_{z,t+1} \tag{28}$$

$$\ln\left(\frac{\mu_{\Upsilon,t+1}}{\mu_\Upsilon}\right) = \rho_\Upsilon \ln\left(\frac{\mu_{\Upsilon,t}}{\mu_\Upsilon}\right) + \epsilon_{\Upsilon,t+1}, \tag{29}$$

where $\epsilon_{z,t+1} \sim \mathcal{NID}\left(0, Var\left(\epsilon_{z,t+1}\right)\right)$ and $\epsilon_{\Upsilon,t+1} \sim \mathcal{NID}\left(0, Var\left(\epsilon_{\Upsilon,t+1}\right)\right)$. We also require that $\rho_z \in]-1,1[$ and $\rho_\Upsilon \in]-1,1[$.

The firms maximize the present discounted value of their dividend payments over $k_{i,t}$, $h_{i,t}$, and $P_{i,t}$. When doing so, they face the following four constraints. The first is the restriction on aggregate demand

$$y_t^d = c_t\left(1 + l\left(v_t\right)\right) + \bar{g}_t + \Upsilon_t^{-1}\left(i_t + a\left(u_t\right)k_t\right), \tag{30}$$

where $\bar{g}_t$ is public spending. The second restriction is a cash-in-advance constraint on a fraction ν of the firms' payments to the workers. Thus, the money demanded by the i'th firm is $m_{i,t}^f = \nu w_t h_{i,t}$. The third constraint is the budget restriction giving rise to the expression for real dividends from firm i in period t, denoted $\phi_{i,t}$

$$\phi_{i,t} = \left(P_{i,t}/P_t\right) y_{i,t}^d - r_t^k k_{i,t} - w_t h_{i,t} - m_{i,t}^f\left(1 - R_{t,1}^{-1}\right) \tag{31}$$

$$-E_t r_{t,t+1} x_{i,t+1}^f + m_{i,t}^f - \pi_t^{-1}\left(x_{i,t}^f + m_{i,t-1}^f\right).$$

The first term in (31) denotes the real revenue from the sale of the i'th good. The firm's expenditures are allocated to: i) purchase of capital services $\left(r_t^k k_{i,t}\right)$, ii) payments to the workers $(w_t h_{i,t})$, and iii) opportunity costs of holding money due to the cash-in-advance constraint $\left(m_{i,t}^f \left(1 - R_{t,1}^{-1}\right)\right)$. The final term in (31) is the change in the firm's real financial wealth.

The fourth constraint introduces staggered price adjustments. We assume that in each period, a fraction $\alpha \in [0, 1[$ of randomly picked firms are not allowed to set the optimal nominal price of the good they produce. Instead, these firms update their prices according to the rule $P_{i,t} = P_{i,t-1}\pi_{t-1}^{\chi}$ where $\chi \in [0, 1]$.

The government: We assume the following process for public spending, denoted $\bar{g}_t$

$$\bar{g}_t = w_t g_t \tag{32}$$

$$\ln\left(\frac{g_{t+1}}{g}\right) = \rho_g \ln\left(\frac{g_t}{g}\right) + \epsilon_{g,t+1}, \tag{33}$$

where $\epsilon_{g,t+1} \sim \mathcal{NID}\left(0, Var\left(\epsilon_{g,t+1}\right)\right)$ and $\rho_g \in]-1, 1[$. Part of public spending is financed by seigniorage and the remaining part by lump-sum taxes.

The central bank: We follow Andreasen (2008*b*) and specify monetary policy by letting

$$\ln R_{t,1} = \ln R_{ss,1} + \alpha_R \ln\left(\frac{R_{t-1,1}}{R_{ss,1}}\right) \tag{34}$$

$$+ \sum_{j=0}^{\infty} d_r^j \left(\alpha_\pi E_t \ln\left(\frac{\pi_{t+1+j}}{\pi_{t+1+j}^*}\right) + \alpha_y E_t \ln\left(\frac{y_{t+1+j}^d / y_{t+j}^d}{\mu_y}\right)\right),$$

where $\alpha_R \in [-1, 1]$, $\alpha_\pi \geq 0$, $\alpha_y \geq 0$ and $d_r \in [0, 1[$. Here, π_t^* is a variable inflation rate target, and μ_y is the growth rate for output in steady state. We assume that

$$\pi_t^* = (1 - \omega^*) \sum_{j=0}^{\infty} (\omega^*)^j \left(\pi_{t-1-j} + \frac{\epsilon_{\pi^*,t-j}}{1 - \omega^*}\right), \tag{35}$$

where $\omega^* \in [0, 1[$ and $\epsilon_{\pi^*,t} \sim \mathcal{NID}\left(0, Var\left(\epsilon_{\pi^*,t}\right)\right)$.

5.2 Determining the structural parameters

The final step in constructing the test economies is to determine the values for the structural parameters. To make our test economies as representative as possible for the DSGE models in the literature, we adopt the following strategy: i) select appropriate sample intervals for each of the structural parameters based on estimation and calibration results in the literature, and ii) draw uniformly from these intervals to generate a total of 100 test economies. The sample intervals are determined based on the results in Christiano et al. (2005), Altig et al. (2005), and Schmitt-Grohé & Uribe (2006) where possible and shown in the table 1.

< Table 1 about here >

As pointed out by Andreasen (2008*a*), certain requirements need to be fulfilled in DSGE models with stochastic and deterministic trends in order to ensure that the households' and the firms' objective functions are finite. Our DSGE model has the same key properties as the model in Andreasen (2008*a*), and hence the results in Andreasen (2008*a*) also apply to our DSGE model. With normally distributed shocks and ϕ_3 not tending to 1, proposition 1(a) in Andreasen (2008*a*) implies the following conditions

$$\exp\left\{\frac{Var\left(\epsilon_{\varepsilon_h,t}\right)}{2\left(1-\rho_{\varepsilon_h}\right)^2}\right\}\beta < 1 \tag{36}$$

$$\exp\left\{\frac{Var\left(\epsilon_{\varepsilon_h,t}\right)}{2\left(1-\rho_{\varepsilon_h}\right)^2}\right\}\exp\left\{\frac{F_\Upsilon^2 Var\left(\epsilon_{\Upsilon,t}\right)}{2\left(1-\rho_\Upsilon\right)^2}\right\}\exp\left\{\frac{F_z^2 Var\left(\epsilon_{z,t}\right)}{2\left(1-\rho_z\right)^2}\right\}\beta\mu_{\Upsilon,ss}^{F_\Upsilon}\mu_{z,ss}^{F_z} < 1 \tag{37}$$

$$F_\Upsilon \equiv \left(1-\phi_4\right)\left(1-\phi_3\right)\frac{\theta}{1-\theta} \tag{38}$$

$$F_z \equiv \left(1-\phi_4\right)\left(1-\phi_3\right) \tag{39}$$

besides a boundedness condition. Andreasen (2008*a*) shows that this boundedness condition is satisfied if we assume that all variables in the economy are never too far away from the economy's growth path. Given this assumption, all our 100 test economies have finite objective functions provided that (36) and (37) hold. We therefore impose (36) and (37) when generating our test economies.

6 The Results

This section studies the performance of the optimization routines. We start by describing the study design which are common for the tests of both optimization routines. Then results are reported for the extended version of Simulated Annealing and the extended CMA-ES routine.

6.1 Study design

All the tests of the optimization routines are carried out based on a first order approximation of the DSGE model. Using a more accurate approximation method would make it impossible to conduct the test of the optimization routines within a reasonable time horizon. However, the first-order approximation is sufficient to determine whether the DSGE model has a unique and stable solution for a particular set of structural parameters and is therefore able to describe the shape of the non-convex parameter space.

Data series for each of the test economies are generated from simulating series of normally distributed shocks and the parameters of the test economies. The length of these series is 200 which corresponds to the typical length of macro series available at a quarterly time frequency. We use the following seven series for our test study: i) the interest rate, ii) the inflation rate, and the real growth rates in iii) GDP, iv) consumption, v) investments, vi) public spending, and vii) labor income. These seven series are referred to as the vector of macro series.

We do not estimate the unobserved states in the DSGE model when evaluating the likelihood function. The argument is that estimation of the states introduces a small error in the likelihood function, and the true values of the structural parameters may not exactly maximize the likelihood function. Thus, if we were to estimate the states it would be impossible to determine whether deviations in the structural parameters from their true values would be due to: i) errors in the estimation of the unobserved states (filtering error) or ii) the failure of the optimization routines to find the global optimum. Therefore, we assume that the initial states and all shocks to the economy are known when evaluating the likelihood function.

Furthermore, we do not include measurement errors in the test data series because simulated measurement errors in samples of 200 observations will never have exactly the same means and variances as the distributions they are generated from. Again, if these measurements were included, the true structural parameters would not exactly maximize the likelihood function.

Given these conditions, maximizing the likelihood function for the test data series corresponds to carrying out Non-linear Least Squares, or equivalently, GMM with the weighting matrix equal to the identity matrix. Hence, the tests of the optimization routines below may also be considered as testing the two routines' ability to minimize the objective function for a moment based estimator.

In order to test the optimization routines on objective functions with increasing dimensionality, we partition the 35 structural parameters in our DSGE model into three groups. The first group consists of the 10 parameters where an A is reported in the second column of table 1. All the remaining parameters are assumed to be known. The second group consists of all the parameters in the first group and the additional 10 parameters where a B is reported in the second column of table 1. All the remaining 15 parameters are assumed to be known. Finally, the third group consists of all 35 parameters. Constructing the three groups of parameters in this way induces a natural progression of increasing difficulty for the optimization routines when testing the routines on the three groups.

6.2 Results for Simulated Annealing

The test of Simulated Annealing is carried out as follows. For each of the 100 test economies, starting values for all unknown parameters are generated by sampling uniformly from the sample intervals in table 1. That is, for each test economy, we generate only one vector of starting values for the optimization routine. We motivate this test procedure by tractability since studying the effect of different starting values would make it impossible to conduct the test within a reasonable amount of time due to the many functions evaluations used by Simulated Annealing.

We let the initial temperature (T) be equal to the squared distance between the true seven macro series and the seven macro series at the starting value for the parameter vector. The sample intervals in table 1 are used as the initial step sizes ($\mathbf{v}$). Preliminary tests showed that these values for T and $\mathbf{v}$ give desirable step sizes at the beginning of the search. For the value of r_T, which controls the reduction rate in the temperature, we consider the performance of Simulated Annealing for $r_T = 0.85$ and for $r_T = 0.90$. The former value is the recommended value in Corana et al. (1987). For the number of random walkers, we test the routine for $N_T = \{10, 20, 30\}$ in the three cases with 10, 20 and 35 unknown parameters in the likelihood function. Thus, our values of N_T are in all cases lower than the values recommended by Corana

et al. (1987) which are $N_T = \max(100, 5n)$ where n is the number of unknown parameters in the likelihood function. We motivate our choices of N_T by tractability of the present study and in future applications because higher values of N_T imply that Simulated Annealing will need too many function evaluations to make the routine of any practical use. For the remaining parameters, we let $N_r = 10$, $N_S = 20$, $N_\epsilon = 4$, $\epsilon_Q = 10^{-6}$, and we impose no upper bound on the number of function evaluations. The search intervals for each of the structural parameters are listed in the final column of table 1

<Table 2 about here >

The results of the tests are reported in table 2. We first notice that the routine does well with 10 unknown structural parameters in the likelihood function ($n = 10$). Here, Simulated Annealing is able to find the global optimum for 85% to 89% of the test economies, depending on the values of r_T and N_T. In all the remaining cases, the algorithm also reports successful termination even though it only finds local optima. Thus, for 11% to 15% of the test economies, Simulated Annealing is trapped in local optima.

With 20 unknown parameters in the likelihood function ($n = 20$), the ability of the routine to find the global optimum is further reduced, in particular for low values of N_T. For $n = 20$, we also find that increasing N_T and/or r_T significantly improves the performance of the routine. However, when all of the structural parameters ($n = 35$) in the DSGE model are considered unknown, the routine is simply unable to find the global optimum for any of the 100 test economies. This breakdown of Simulated Annealing is surprising, and it shows that a larger value of N_T and/or r_T is needed with 35 unknown structural parameters in the likelihood functions.

Finally, we note from table 3 that Simulated Annealing requires a lot of function evaluations and that this number increases rapidly with the number of unknown parameters in the likelihood function.

6.3 Results for CMA-ES routine

The test of the CMA-ES routine is carried out as follows. For each of the 100 test economies, we generate starting values for the unknown structural parameters by sampling uniformly from the sample intervals in table 1. For each test economy, 10 different starting values are generated. Thus, the test of the CMA-ES routine is carried out in a slightly different way than the test of Simulated Annealing. The reason being that the CMA-ES algorithm uses notably fewer function evaluations than Simulated Annealing, and this fact makes it possible to do an extended test of the CMA-ES routine. Hence, in testing the CMA-ES routine, we perform a total of $10 \times 100 = 1000$ optimizations. In scheme $\mathcal{A}$, we report the results of all 1000 optimizations. However, our setup also allows for the construction of a scheme $\mathcal{B}$ where only the best value for a given test economy is reported. That is, we use the CMA-ES routine 10 times with 10 different starting values on the same test economy, and based on these 10 attempts to optimize the likelihood function, we only report the best value. Thus, scheme $\mathcal{B}$ is very similar to the test done by Auger & Hansen (2005) which introduces a restarting procedure in relation to the CMA-ES routine. However, the procedure in Auger & Hansen (2005) deviates from our procedure because they gradually increase the population size in each restart and make the number of restarts endogenous.

We test the performance of the CMA-ES routine for different sizes of the population parameter (λ). The values for the function tolerance and the tolerance in the unknown parameters for the objective function are set equal to 10^{-6}. All the other parameters in the routine are at their default values, and we allow for an unlimited number of function evaluations and an unlimited number of iterations.

< Table 3 about here >

Table 3 reports the results for the CMA-ES routine with 10 unknown structural parameters in the likelihood function. In scheme $\mathcal{A}$, we see that the routine is able to optimize between 78% and 88% of the likelihood functions. Hence, the performance of the CMA-ES routine corresponds to the performance of Simulated Annealing for $n = 10$. But, the CMA-ES routine achieves the same success rate with notably fewer function evaluations, only about $20,000$ per optimization. This difference in the number of function evaluations among the two routines is approximately a factor of 30, meaning that the CMA-ES routine can optimize 30 likelihood functions for every likelihood function Simulated Annealing optimizes. We also note that increasing the population size (λ) increases the probability of finding the global optimum and reduces the probability of ending up in local optima. Both results are in line with the theory for the CMA-ES algorithm as described above.

In scheme $\mathcal{B}$, we first notice that the probability of finding the global optimum is even higher than in scheme $\mathcal{A}$. Now, the routine is able to optimize between 91% and 95% of the test economies. Moreover, the probability of ending up in local optima is very low, and with a population size of 250, the risk of local optima is completely eliminated. However, this improved performance of the CMA-ES routine comes at the cost of using 10 times more function evaluations than in scheme $\mathcal{A}$. Still, the number of function evaluations used in scheme $\mathcal{B}$ is considerably lower than the number of function evaluations used in Simulated Annealing.

< Table 4 about here >

The results with 20 unknown parameters in the likelihood function are reported in table 4. In scheme $\mathcal{A}$, we find that the CMA-ES routine can optimize between 66% and 75% of the likelihood functions, and the probability of only finding a local optima is higher than with 10 unknown parameters. Again, increasing the size of the population (λ) improves the performance of the routine at the cost of more function evaluations. Table 5 also shows that the CMA-ES routine clearly outperforms Simulated Annealing in its ability to optimize the likelihood functions and in efficiency. The CMA-ES algorithm uses only about $70,000$ function evaluations per optimization. Again, the difference in the number of function evaluations is approximately a factor of 30.

In scheme $\mathcal{B}$ with 20 unknown parameters in the likelihood function, we find that the CMA-ES routine is able to optimize between 77% and 85% of the test economies. Moreover, the probability of ending up in local optima may be reduced to between 4% and 6% of the test economies by setting $\lambda \geq 100$.

< Table 5 about here >

In scheme $\mathcal{A}$ with 35 unknown parameters, we see that the CMA-ES routine has more trouble optimizing the likelihood functions, particularly for population sizes less than 50. Increasing the population size to between 150 and 250 improves the performance of the algorithm considerably,

and the routine is then able to optimize between 47% and 58% of the likelihood functions, respectively. However, this improved performance comes at the cost of using on average between $200,000$ and $250,000$ function evaluations. Still, the CMA-ES routine is much more efficient than Simulated Annealing and clearly outperforms Simulated Annealing in its ability to optimize the likelihood functions.

Going from scheme $\mathcal{A}$ to scheme $\mathcal{B}$, we observe a large improvement in the performance of the CMA-ES routine. With a population size of 100, the routine is now able to optimize 71% of the test economies, and only in 5% of the test economies does the CMA-ES routine report local optima.

A final remark should be made about the CMA-ES routine. In all three test cases, i.e. for $n = \{10, 20, 35\}$, the same pattern appears: increasing the population size from 250 to 500 reduces the routine's ability to optimize the likelihood functions. To our knowledge, this is a new result in relation to the CMA-ES routine. However, it is beyond the scope of this paper to derive optimal values for the population size. But, as a rule of thump we recommend that the population size in schemes $\mathcal{A}$ and $\mathcal{B}$ should be at least three times larger than the number of unknown parameters in the likelihood function ($\lambda \geq 3n$). Using this rule, the researcher has a good chance of finding the global optimum and only faces a low risk of ending up in local optima.

6.4 An overall evaluation of the two optimization routines

This section conducts an overall evaluation of the two optimization routines. The first interesting question in relation to Simulated Annealing and the CMA-ES routine is whether they outperform a standard optimization routine. We examine this question by testing the performance of the well-known simplex algorithm by Nelder and Mead on the 100 test economies. Points where the likelihood function is not defined are assigned a very large value. This generates a convex search space but also strong kinks in the objective function. However, the Nelder-Mead simplex algorithm only relies on function evaluations and is thus able to handle these kinks in the objective function. This fact is our motivation for choosing the Nelder-Mead simplex algorithm in advance of derivative based optimization routines. We adopt the same testing strategy for the Nelder-Mead simplex routine as for the CMA-ES routine because the Nelder-Mead routine uses relatively few function evaluations.[8]

<Table 6 about here>

The results for the Nelder-Mead simplex routine are reported in table 6. With 10 unknown parameters in the likelihood function, the Nelder-Mead routine is able to find the global optimum for 27.7% of the likelihood functions in scheme $\mathcal{A}$. The corresponding number is 66% in scheme $\mathcal{B}$. However, the routine is not able to find any of the global optima with 20 or 35 unknown parameters in the likelihood function. Moreover, the number of times the routine reports a local minima is very high. In scheme $\mathcal{A}$, this number is 85% and 95% for 20 and 35 unknown parameters in the likelihood function, respectively. The corresponding numbers in scheme $\mathcal{B}$ are 98% and 99%.

[8] We use the version of the Nelder-Mead routine implemented in Matlab by the routine fminsearch. We refer to Lagarias, Reeds, Wright & Wright (1998) for a description of the Nelder-Mead algorithm.

The number of function evaluations reported in table 6 must be interpreted with great care. The reason being that we impose an upper bound of $300,000$ function evaluations and an upper bound of $90,000$ iterations when using the Nelder-Mead routine. If we do not impose these restrictions, the routine will in some cases continue the search for a very long time without generating any significant improvement in the value of the likelihood function. Hence, we impose these upper bounds to keep the test study tractable and not generate a more or less arbitrarily large number of function evaluations for the routine. We emphasize that the objective functions are very far from the global optimum in all cases where the routine terminates due to reaching one of these upper bounds.

Summarizing, Simulated Annealing and, in particular, the CMA-ES routine clearly outperform the Nelder-Mead routine in terms of being able to find the global optimum for our test economies.

A second interesting question is whether the researcher can learn anything besides the true optimum from using Simulated Annealing or the CMA-ES routine? This is indeed the case because both routines report the variances in the search distributions for each of the unknown parameters during the optimization process. This helps the researcher to determine whether one or more structural parameters in the DSGE model are weakly or entirely unidentified even before the global optimum is found. That is, if the likelihood function is almost flat along a given dimension, then the variance in the search distribution along this dimension will be very large. Hence, a researcher does not have to go through a potentially very troubling process of optimizing a likelihood function which has almost flat segments because such features are easily revealed during the optimization process. In this case, the researcher may choose to: i) respecify the model, ii) include more data in the estimation, and/or iii) determine the values of some parameters by calibration arguments.

Having established that Simulated Annealing and the CMA-ES routine clearly outperform a standard optimization routine and both provide the researcher with valuable information during the optimization, we ask the question: which of the two optimization routines do we prefer? At least four arguments are in favor of the CMA-ES routine. First, it clearly outperforms Simulated Annealing in its ability to find the global optimum. Second, the CMA-ES routine is much more efficient than Simulated Annealing. Third, Simulated Annealing has at least three tuning parameters $(N_T, r_T, T,)$, whereas the CMA-ES routine has only one, the population size (λ). Fourth, the CMA-ES routine is very easy to implement in a multiprocessing framework, i.e. in a setting where two or more cpu's work on the same problem. The latter is an important property because the use of multiprocessing can reduce the physical time of an optimization. Multiprocessing in the CMA-ES routine is normally implemented by letting up to λ cpu's evaluate one or more of the required function evaluations needed in each iteration. Besides being an easy way to implement multiprocessing, it is also a very efficient way because the amount of communication between the cpu's is very low. Hansen et al. (2003) use this procedure, and they show that the speed-up in the CMA-ES routine scales almost linearly with the number of cpu's. On the other hand, the Simulated Annealing algorithm is intrinsic sequential and can not directly be implemented in a multiprocessing framework.

7 Conclusion

The contribution of this paper is twofold. It extends two well-known optimization routines to deal with objective functions for DSGE models and it tests their ability to optimize likelihood functions for a DSGE model. Our test results show that the extended CMA-ES routine clearly outperforms the extended version of Simulated Annealing. That is, the CMA-ES routine finds the global optimum more often than Simulated Annealing does, and the CMA-ES routine uses notably fewer function evaluations than Simulated Annealing. Hence, the CMA-ES routine is the most efficient optimization routine of the two. Second, with 10 unknown structural parameters in the likelihood function, the CMA-ES routine finds the global optimum in 95% of our test economies compared to 89% for Simulated Annealing. When the number of unknown structural parameters in the likelihood function increases to 20 and 35, then the CMA-ES routine finds the global optimum in 85% and 71% of the test economies, respectively. The corresponding numbers for Simulated Annealing are 70% and 0%.

We conjecture that the good performance of the CMA-ES routine is primarily due to the inclusion of co-variance terms in the search distribution. Hence, future research could focus on how to introduce covariances in the search distribution for the random walker in Simulated Annealing. One way of doing this could be to sample from a multivariate normal distribution when generating moves for the random walker, and then update the covariance matrix $\mathbf{C}$ in a similar manner as done in the CMA-ES routine. However, a correction for autocorrelation in the random walker's sample path must be included in the update of the covariance matrix. Still, the Metropolis criterion should be applied in this new version of Simulated Annealing to decide whether a suggested move should be accepted or rejected. Finally, the global step size could be reduced gradually like the temperature in the version of Simulated Annealing presented above.

Future research could also focus on deriving the optimal population size (λ) in the CMA-ES routine or examine whether the efficiency of the routine can be improved by using the restart strategy suggested in Auger & Hansen (2005).

Figure 1: Pseudo-codes for one iteration in Simulated Annealing

```
FOR m = 1:N_T
    FOR j = 1:N_S
        FOR h = 1:n
            "Draw from the uniform distribution for dimension h"
            x(h)' = x(h) + v(h)u(h) and x(i)' = x(i) for i ≠ h

            IF ( x(h)' ∉ [a(h),b(h)] ) THEN
                draw x(h)' from Uni(a(h),b(h))
            END IF

            "Accept or reject the move according to the Metropolis Criterion"
            IF ( min{exp{-ΔQ/T},1}> Uni(0,1) )  THEN
                accept the move
            ELSE
                reject the move
            END
        END
    END
    "Adjust the variances for all n search distributions"
END
```

Table 1: Parameters for the test economies

The fourth column named "Sample" denotes the intervals we sample from when generating the test economies. The fifth column named "Search" denotes the intervals we search in during the optimizations.

			Interval	
Label	Group	The households	Sample	Search
Discount factor	B	β	$[0.98, 1[$	$[0, 1[$
Habit degree	A	b	$[0.2, 0.8]$	$[0, 1]$
Transaction cost	B	ϕ_1	$[0.05, 0.1]$	$[0, 100]$
Transaction cost		ϕ_2	$[0.1, 0.2]$	$[0, 100]$
Preference	A	ϕ_3	$[1, 3]$	$[0, 100]$
Preference		ϕ_4	$[0.3, 0.7]$	$]0, 1[$
Adj costs for invest	A	κ	$[2, 4]$	$[0, 100]$
Cost of utilization		γ_2	$[0.05, 0.1]$	$[0, 100]$
Wage elasticity		$\tilde{\eta}$	$[15, 35]$	$]1, 100]$
Degree of wage stickiness	A	$\tilde{\alpha}$	$[0.3, 0.7]$	$[0, 1[$
Wage Indexation	B	$\tilde{\chi}$	$[0, 1]$	$[0, 1]$
labor supply		h_{ss}	$[0.2, 0.8]$	$[0, 1]$
		The firms		
Depreciation rate	B	δ	$[0.02, 0.04]$	$[0, 1]$
Cobb-Douglas parameter	A	θ	$[0.28, 0.38]$	$]0, 1[$
CIA		ν	$[0, 1]$	$[0, 1]$
Price elasticity	B	η	$[4, 12]$	$]1, 100]$
Degree of price stickiness	B	α	$[0.5, 0.9]$	$[0, 1[$
Price Indexation		χ	$[0, 1]$	$]0, 1[$
		The central bank		
Process for inflation target	A	ω^*	$[0.5, 0.9]$	$[0, 1[$
Reaction to lagged interest rate	B	α_R	$[-1, 1]$	$[-1, 1]$
Reaction to inflation		α_π	$[0, 2]$	$[0, 100]$
Reaction to output	A	α_y	$[1, 5]$	$[0, 100]$
Forward looking-ness		d_r	$[0, 1]$	$[0, 1[$
Inflation rate in steady state		π_{ss}	$[0.0098, 0.0102]$	$[0, 100]$

Table 1: Parameters for the test economies (continued)

			Interval	
Label	Group	Ex. processes	Sample	Search
Persistency in neutral tech	A	ρ_z	$[0, 0.8]$	$]-1, 1[$
Persistency in embodied tech	B	ρ_Υ	$[0, 0.3]$	$]-1, 1[$
Persistency in preference shocks		ρ_{e_h}	$[0, 0.3]$	$]-1, 1[$
Persistency in government shocks		ρ_g	$[0.6, 1)$	$]-1, 1[$
Std for shocks to neutral tech		$\sqrt{Var\left(\epsilon_{z,t+1}\right)}$	$[0.0005, 0.001]$	$[0, 100]$
Std for shocks to embodied tech	B	$\sqrt{Var\left(\epsilon_{\Upsilon,t+1}\right)}$	$[0.003, 0.005]$	$[0, 100]$
Std for shocks to preferences	A	$\sqrt{Var\left(\epsilon_{\mu_{\varepsilon_h},t+1}\right)}$	$[0.001, 0.005]$	$[0, 100]$
Std for shocks to government spendings		$\sqrt{Var\left(\epsilon_{g,t+1}\right)}$	$[0.005, 0.01]$	$[0, 100]$
Std for shocks to inflation rate target		$\sqrt{Var\left(\epsilon_{\pi^*,t+1}\right)}$	$[0.05, 0.1]$	$[0, 100]$
Growth rate for neutral tech	A	μ_z	$[1, 1.005]$	$[1, 100]$
Growth rate for embodied tech	B	μ_Υ	$[1, 1.005]$	$[1, 100]$

Table 2: Results for the Simulated Annealing

This table shows the percentage of times the routine finds the global optimum or ends up in a local optimum based on 100 test economies. The table also shows the average number of function evaluations used during an optimization. The criterion for finding the true optimum is that the function value returned from the routine is less than 10^{-6} from the true optimum. The initial value for the temperature (T) is equal to the squared distance between the true macro series and the macro series at the starting point for the optimization routine. The other parameters for the routine are given the values: $N_S = 20$, $N_\epsilon = 4$, $\epsilon_Q = 10^{-6}$ and $N_r = 10$.

		r_T= 0.85			r_T= 0.90		
	N_T	found opt	local opt	Avg number of fct evaluations	found opt	local opt	Avg number of fct evaluations
$n = 10$	10	85%	15%	344,891	85%	15%	510,551
	20	85%	15%	645,042	87%	13%	959,629
	30	88%	12%	939,806	89%	11%	1432,099
$n = 20$	10	34%	66%	880,306	43%	57%	1,168,524
	20	58%	42%	1,470,252	66%	34%	2,178,387
	30	67%	33%	2,087,413	70%	30%	3,118,636
$n = 35$	10	0%	100%	2,045,942	0%	100%	2,532,779
	20	0%	100%	3,105,162	0%	100%	4,645,399
	30	0%	100%	4,641,859	0%	100%	6,570,895

$n =$ the number of unknown parameters in the likelihood function

$N_T =$ the number of random walkers

$r_T =$ the reduction rate in temperature

Table 3: Results for the CMA-ES routine, n = 10

The criterion for finding the true optimum is that the function value returned from the CMA-ES is less than 10^{-6} from the true optimum. A value of 10 for the population size in the CMA-ES routine is the lowest recommended value. All other parameters in the CMA-ES routine are at their default values as specified in section 4.1.

	Scheme $\mathcal{A}$			Scheme $\mathcal{B}$			
Population	Pct. of all the 1000 runs where:			Pct.of economies based on 10 attempts where:			Avg. number of function
size (λ)	found opt	no convergence	local opt	found opt.	no convergence	local opt	evaluations for 1 opt.
10	77.6	4.6	17.8	92	2	6	8,678
20	81.3	5.8	12.9	93	2	5	11,179
30	83.6	5.6	10.8	94	3	3	11,427
40	83.9	7.0	9.1	94	5	1	12,123
50	84.5	7.9	7.6	94	4	2	16,961
80	86.7	7.5	5.8	94	4	2	18,625
100	86.0	8.3	5.7	95	4	1	24,579
150	87.3	9.1	3.6	93	5	2	21,898
200	87.7	9.4	2.9	93	3	4	23,271
250	88.5	9.5	2.0	93	7	0	28,426
500	88.1	10.8	1.1	91	8	1	41,236

Table 4: Results for the CMA-ES routine, n = 20

The criterion for finding the true optimum is that the function value returned from the CMA-ES is less than 10^{-6} from the true optimum. A value of 12 for the Population size in the CMA-ES routine is the lowest recommended value. All other parameters in the CMA-ES routine are at their default values as specified in section 4.1.

	Scheme $\mathcal{A}$			Scheme $\mathcal{B}$			
Population	Pct. of all the 1000 runs where:			Pct.of economies based on 10 attempts where:			Avg. number of function
size (λ)	found opt	no convergence	local opt.	found opt.	no convergence	local opt	evaluations for 1 opt.
12	66.2	12.2	21.6	85	1	14	126,213
20	69.7	14.0	16.3	85	4	11	128,178
30	70.1	13.7	16.2	84	5	11	69,365
40	70.0	14.5	15.5	85	9	6	63,246
50	71.7	16.3	13.0	84	7	9	69,032
80	71.0	18.7	10.3	82	10	8	64,187
100	72.2	18.3	9.5	83	11	6	70,304
150	72.5	17.8	9.7	80	14	6	73,823
200	75.1	16.8	8.1	81	13	6	77,567
250	74.1	17.3	8.6	80	14	6	89,282
500	71.8	23.4	4.8	77	19	4	134,238

Table 5: Results for the CMA-ES routine, n = 35

The criterion for finding the true optimum is that the function value returned from the CMA-ES is less than 10^{-6} from the true optimum. A value of 14 for the Population size in the CMA-ES routine is the lowest recommended value. All other parameters in the CMA-ES routine are at their default values as specified in section 4.1.

	Scheme $\mathcal{A}$			Scheme $\mathcal{B}$			
Population	Pct. of all the 1000 runs where:			Pct.of economies based on 10 attempts where:			Avg. number of function
size (λ)	found opt	no convergence	local opt.	found opt.	no convergence	local opt	evaluations for 1 opt.
14	7.9	22.9	69.2	33	3	64	161,420
20	11.6	23.7	64.7	44	4	52	159,915
30	13.8	26.1	60.1	46	12	42	161,585
40	14.7	27.1	58.2	47	17	36	163,215
50	20.6	29.6	49.8	56	22	22	165,006
80	30.4	31.0	38.6	66	25	9	179,137
100	38.1	31.8	30.1	71	24	5	192,819
150	47.2	33.1	19.7	70	27	3	215,882
200	55.8	35.5	8.7	69	28	3	241,897
250	57.9	36.6	5.5	70	29	1	260,827
500	45.1	42.1	12.8	57	34	9	358,247

Table 6: Results for the Nelder-Mead Simplex Algorithm

The criterion for finding the true optimum is that the function value returned from the Nelder-Mead Simplex Algorithm is less than 10^{-6} from the true optimum. An upper bound of 300,000 function evaluations and an upper bound of 90,000 iterations are imposed.

	Scheme $\mathcal{A}$			Scheme $\mathcal{B}$			
	Pct. of all the 1000 runs where:			Pct.of economies based on 10 attempts where:			Avg. number of function
n	found opt	no convergence	local opt	found opt.	no convergence	local opt	evaluations for 1 opt.
10	27.7	5	67.3	66	1	33	18,262
20	0	5.4	94.6	0	1	99	30,682
35	0	15.3	84.7	0	2	98	80,699

References

Altig, D., Christiano, L. J., Eichenbaum, M. & Linde, J. (2005), 'Firm-specific capital, nominal rigidities and the business cycle', *NBER Working Paper* **no 11034**, 1–50.

An, S. & Schorfheide, F. (2007), 'Bayesian analysis of DSGE models', *Econometric Review* .

Andreasen, M. M. (2008*a*), 'Ensuring the validity of the micro foundation in DSGE models', *Working Paper* .

Andreasen, M. M. (2008*b*), 'A nonlinear micro founded term structure model with macro variables', *Working Paper* .

Auger, A. & Hansen, N. (2005), 'A restart CMA evolution strategy with increasing population size', *Proceedings of the IEEE Congress on Evolutionary Computation, CEC* pp. 1769–1776.

Christiano, L. J., Eichenbaum, M. & Evans, C. L. (2005), 'Nominal rigidities and the dynamic effects of a shock to monetary policy', *Journal of Political Economy* **113**, 1–45.

Corana, A., Marchesi, M. & Ridella, S. (1987), 'Minimizing multimodal functions of continuous variables with "simulated annealing" algorithm', *ACM Transactions on Mathematical Software* **13, No. 3**, 262–280.

Fernández-Villaverde, J. & Rubio-Ramírez, J. F. (2007), 'Estimating macroeconomic models: A likelihood approach', *Review of Economic Studies* pp. 1–46.

Goffe, W. L., Ferrier, G. D. & Rogers, J. (1994), 'Global optimization of statistical functions with simulated annealing', *Journal of Econometrics* **60**, 65–99.

Hansen, N. (2005), 'The CMA evolution strategy: A tutorial', *Working Paper* .

Hansen, N. & Kern, S. (2004), 'Evaluating the CMA evolution strategy on multimodal test functions', *Springer-Verlag: Eighth International Conference on Parallel Problem from Nature* **PPSN VIII**, 282–291.

Hansen, N., Müller, S. D. & Koumoutsakos, P. (2003), 'Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)', *Evolutionary Computation* **11, Number 1**.

Justiniano, A. & Primiceri, G. E. (2008), 'The time varying volatility of macroeconomic fluctuations', *The American Economic Review* pp. 604–641.

Kern, S., Muller, S. D., Hansen, N., Buche, D., Ocenasek, J. & Koumoutsakos, P. (2004), 'Learning probability distributions in continuous evolutionary algorithms - a comparative riview', *Natural Computing* **3(1)**, 77–112.

Lagarias, J. C., Reeds, J. A., Wright, M. H. & Wright, P. E. (1998), 'Convergence properties of the nelder-mead simplex method in low dimensions', *SIAM Journal of Optimization* **9 number 1**, 112–147.

Muller, S. D., Hansen, N. & Koumoutsakos, P. (2002), 'Increasing the serial and the parallel performance of the CMA-evolution strategy with large populations', *Parallel Problem Solving from Nature* .

Rossi, G. D. (2004), 'Maximum likelihood estimation of the cox-ingersoll-ross model using particle filters', *Working Paper* .

Salamon, P., Sibani, P. & Frost, R. (2002), *Facts, Conjectures and Improvements for Simulated Annealing*, Society for Industrial and Applied Mathematics, Philadelphia.

Schmitt-Grohé, S. & Uribe, M. (2006), 'Optimal inflation stabilization in a medium-scale macroeconomic model', *Working Paper* pp. 1–59.

Smets, F. & Wouters, R. (2003), 'Shocks and frictions in US business cycles: A bayesian DSGE approach', *Working Paper* pp. 1–27.

Research Papers 2008

2008-19: Anne Péguin-Feissolle, Birgit Strikholm and Timo Teräsvirta: Testing the Granger noncausality hypothesis in stationary nonlinear models of unknown functional form

2008-20: Stefan Holst Bache, Christian M. Dahl and Johannes Tang Kristensen: Determinants of Birthweight Outcomes: Quantile Regressions Based on Panel Data

2008-21: Ole E. Barndorff-Nielsen, José Manuel Corcuera, Mark Podolskij and Jeannette H.C. Woerner: Bipower variation for Gaussian processes with stationary increments

2008-22: Mark Podolskij and Daniel Ziggel: A Range-Based Test for the Parametric Form of the Volatility in Diffusion Models

2008-23: Silja Kinnebrock and Mark Podolskij: An Econometric Analysis of Modulated Realised Covariance, Regression and Correlation in Noisy Diffusion Models

2008-24: Matias D. Cattaneo, Richard K. Crump and Michael Jansson: Small Bandwidth Asymptotics for Density-Weighted Average Derivatives

2008-25: Mark Podolskij and Mathias Vetter: Bipower-type estimation in a noisy diffusion setting

2008-26: Martin Møller Andreasen: Ensuring the Validity of the Micro Foundation in DSGE Models

2008-27: Tom Engsted and Thomas Q. Pedersen: Return predictability and intertemporal asset allocation: Evidence from a bias-adjusted VAR model

2008-28: Frank S. Nielsen: Local polynomial Whittle estimation covering non-stationary fractional processes

2008-29: Per Frederiksen, Frank S. Nielsen and Morten Ørregaard Nielsen: Local polynomial Whittle estimation of perturbed fractional processes

2008-30: Mika Meitz and Pentti Saikkonen: Parameter estimation in nonlinear AR-GARCH models

2008-31: Ingmar Nolte and Valeri Voev: Estimating High-Frequency Based (Co-) Variances: A Unified Approach

2008-32: Martin Møller Andreasen: How to Maximize the Likelihood Function for a DSGE Model