Efficient calibration of a financial agent-based model using the method of simulated moments

Piotr Zegadło$^{[0000-0002-0219-297X]}$

Kozminski University, Warsaw, Poland
pzegadlo@kozminski.edu.pl

Abstract. We propose a new efficient method of calibrating agent-based models using the Method of Simulated Moments. It utilizes the Filtered Neighborhoods optimization algorithm, gradually narrowing down the search area by examining local neighborhoods of promising solutions. The new method obtains better calibration accuracy for a benchmark financial agent-based model in comparison to a broad selection of other methods, while using just a tiny fraction of their computational budget.

Keywords: Agent-Based Models · Heterogeneous Agent Models · Model Calibration

1 Introduction

Agent-based models (ABMs) have been extensively used to investigate the rich dynamics of financial time series. They can easily incorporate investor heterogeneity and bounded rationality, replicating the most important stylized facts of the financial markets as a result [10]. Such models have been also found to assist in asset pricing [13]. The vast body of literature on this topic can be generally divided into two main strands. One involves large, multi-agent computational models emulating financial market structure [20]. The other focuses on simpler models, with usually only a few groups of interacting agents guided by certain behavioral rules [14]. Such models, sometimes referred to as heterogeneous agent models (HAMs), sacrifice a certain amount of flexibility and complexity for the sake of greater interpretability and an insight into the underlying mechanism of market price dynamics [8].

Although many theoretical HAMs have been shown to be useful in analyzing financial market dynamics, they have also come under criticism for their apparent subjectivity. The need to reconcile theoretical underpinnings with empirical observations has spurred the growth of HAM calibration and validation literature in the recent years [21]. In that context, calibration can be defined as matching the output of a HAM employing a specific set of parameter values to the characteristics of a reference dataset[1]. The quality of that match and

[1] The notions of "calibration" and "validation" of HAMs do not have unambiguous definitions. Here they are used broadly, designating a process of arriving at HAM parameter values that provide the best fit of model output to some reference series.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_27

the parameter values can then be taken under scrutiny, potentially vindicating a specific model – or, conversely, suggesting its modification or abandonment altogether.

A canon of best practices for agent-based models' calibration has not yet been firmly established. This is partially because the modern models are often presented in the form of an analytically intractable set of stochastic difference equations. This spurs most researchers tackling the task of finding the parameters of such models to take the approach of numerical optimization based on a certain heuristic. The Method of Simulated Moments (MSM) is widespread, where parameter calibration is performed through minimizing the distance between a set of statistical moments for the actual observations and the simulated series [7].

Due to the complex dynamics inherent in HAMs, assuring the convergence of a distance metric to the global minimum is difficult. It necessitates evaluating the objective function numerous times through simulations, which may generate a substantial computational cost [22]. What is more, even finding the global minimum of a loss function may not bring the correct parameter set values, due to the stochastic component present in typical financial HAMs. Figure 1 exemplifies this problem using two stylized loss functions - noiseless and noisy. The global minimum at $X = 1000$ is clearly visible for the noiseless function. After adding noise to the loss function, the actual minimum appears in the vicinity of $X = 750$, although we would prefer our optimizer to still uncover $X = 1000$ as the optimum here. The fact of the loss function being noisy should clearly be taken into account in the HAM calibration process.

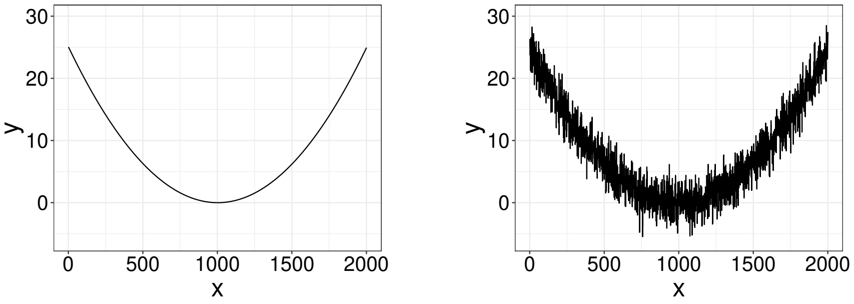

Fig. 1. Example of stylized noiseless (LHS) and noisy (RHS) loss functions.

This work proposes a new algorithm for calibrating HAM parameter values within the Method of Simulated Moments paradigm, called Filtered Neighborhoods. The algorithm evaluates the distance function for different parameter sets chosen with random search. The search area is gradually narrowed down through the examination of the local neighborhoods of promising solutions. Parameter sets associated with large errors are filtered out from the process. In each iteration, new model calls are allotted to searching the areas picked with the help of an acquisition function often employed in Bayesian optimization.

The proposed Filtered Neighborhoods method is applied to calibrate a classic benchmark heterogeneous agent model [4]. Referencing the broad set of methods tested by [22] to solve the same task, it is shown that the new algorithm achieves better results than several competing HAM calibration methods, employing at most ca. 0.2% of the model calls allotted for the competition. The algorithm allows the usage of any distance metrics and objective functions. It can potentially be used for other noisy optimization problems.

2 Construction of the Proposed Calibration Approach

2.1 The Method of Simulated Moments

The family of methods applied in the calibration of agent-based models is broad. Recent surveys suggest different breakdowns of this family into multiple groups [11, 21, 22]. The Method of Simulated Moments is nevertheless prominently featured in the literature, as shown by [7]. It is also considered to be the most straightforward way for calibrating complex models [21].

MSM is a simulation-based method, where first a set of statistical moments for the actual data is picked. For financial HAMs, that actual data is typically a vector (time series) of prices or returns. Then multiple runs of the time series are simulated within the HAM, using different sets of parameter values. The resulting moments for the simulated data are then compared against the moments of the actual dataset. The parameter set best calibrated to the real data is chosen based on the value of the distance function between the moments of the simulated and the real samples.

However, the rather intuitive concept of MSM is not without its issues. Let us consider the following general representation of MSM, where we assume that the reference data at our disposal was generated by a process described perfectly by a given Agent-Based Model:

$$\hat{\theta} = arg \left\{ min \, \mathcal{D} \left(f(\theta, \varepsilon_r), f(\hat{\theta}, \varepsilon_s) \right) \right\} \tag{1}$$

where $\hat{\theta}$ denotes the calibrated set of parameter values based on the simulations and θ is the actual (unobservable) set of parameter values for the reference data. The value of function f is the vector of moments to be used in the calibration process. In accordance with the specification of the ABM being calibrated, function f takes a set of parameter values as an argument, alongside a noise term ε_r for the reference data and ε_s for the simulations. This notation clearly shows that even perfectly matching $\hat{\theta}$ to θ may not drive down the value of the distance function $\mathcal{D}$ to 0. What is more, it may not result in achieving the global minimum of the distance function for a given realization of the noise term. This stems from the fact that $\mathcal{D} \left(f(\theta, \varepsilon_r), f(\hat{\theta}, \varepsilon_s) \right)$ is essentially a random variable with a (usually) unknown distribution. Conversely, finding a global minimum

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_27

of the distance function does not guarantee minimizing the fitness function $\mathcal{F}$ which measures the distance between the true and the calibrated parameter set.

This dichotomy of working with two objective functions in the process of noisy calibration needs to be stressed to avoid confusion. In this paper, the distance function $\mathcal{D}$ denotes the distance between the true set of moments and the simulated set of moments. Its values are observable and it serves as the driver of the calibration process. However, the final goodness of fit of the calibration is evaluated using the fitness function $\mathcal{F}$, gauging the distance between the calibrated set of parameters and the true set of parameters. The actual goal in the process is to minimize the value of $\mathcal{F}$, and not necessarily $\mathcal{D}$. Importantly, in the simulation exercises, the true set of parameters is known, which enables calculating the actual value of $\mathcal{F}$. In empirical work, the value of the fitness function remains unobservable.

Chen, Chang and Du [6] remarked that this indirect inference about θ (using the observable $\mathcal{D}$ in place of the unobservable $\mathcal{F}$) can be impacted by a number of decisions a researcher has to make, including:

- the number of the moments,
- the selection of the moments,
- the (sub)set of model parameters being calibrated,
- the form of the distance function $\mathcal{D}$,
- the optimization algorithm used to arrive at $\hat{\theta}$,
- the computational budget, i.e. the number of model calls available during simulations.

This broad degree of subjectivity remains the main cause for criticism of MSM [11, 22]. Therefore in this work we concentrate on developing just a single element of the process - the optimization algorithm for searching of $\hat{\theta}$ - considered to be an important research topic [21]. Other design choices for the MSM specification used in this paper will be made in accordance with common choices made in the literature.

2.2 Details of the Filtered Neighborhoods optimization algorithm

The proposed algorithm entails a modified random search procedure through a gradually narrowed search space. The narrowing is performed with the help of an acquisition function typically used in Bayesian optimization.

Random search is a widespread approach to hyper-parameter optimization in machine learning [2]. It can be applicable in this case, too, as loss functions for both machine learning models and HAMs are essentially nonlinear, noisy functions of multiple parameters. The first step of the proposed algorithm entails drawing an initial sample of n HAM parameter values to obtain an initial position for the later search. In order to ensure even sampling across the dimensions of the parameter space, quasi-random low-discrepancy Sobol sequences with Owen and Faure-Tezuka scrambling are employed [9, 15]. Bergstra and Bengio [2] demonstrated that the use of the Sobol sequence in random search can be

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_27

advantageous, especially if the generated number of observation points amounts to several hundred.

The second step of the algorithm filters out all parameter value sets with a high value of the distance function $\mathcal{D}$, leaving only β points with the lowest values of the distance metric for further examination. This is done for two reasons. Firstly, the noisiness of the objective function may be very complex. The relationship between specific parameter values and the distance metric can vary with the value of the objective function. As the goal of the optimization process is searching for minima, that relationship is crucial above all for the low values of the objective function. Secondly, limiting the sample to the points which have the potential of having the most informational value can be beneficial for computational time, without adversely affecting the results [16].

The third step entails picking the most promising areas of the parameter space in order to perform the next batch of simulations concentrated on such areas. Acquisition functions such as probability of improvement, expected improvement or lower confidence bound are often employed for a similar purpose in Bayesian optimization [25]. However, typically they are set up to pick only a single point from the parameter space before proceeding with the simulation. Also, the process depends strongly on the chosen prior [3]. Instead of assuming a prior distribution, we find α nearest neighbors of each set of parameter values in the sample[2] (including itself). In this way, β neighborhoods including α points each are created. Such local neighborhoods can then be characterized by the mean value of the distance metric and its standard error. Minimum and maximum parameter values for each of the points constitute the borders of the neighborhood's area. This setup allows the use of an acquisition function to rank the neighborhoods from the most to the least promising in the fourth step of the algorithm. Using a portfolio of multiple acquisition functions can also be beneficial [24]. It is not considered in this paper in order to keep the number of parameters in the algorithm as low as possible. For the centroid of the single most promising neighborhood in each optimization round, the value of the distance function $\mathcal{D}$ is recorded (a single additional model call per round).

Then, in the fifth step, the budget of n simulations is distributed among γ most promising neighborhoods, resulting in n/γ additional model calls per neighborhood.

The sixth step of the algorithm entails randomly picking the allocated number of new parameter set values for each of the neighborhoods within the designated borders, using the continuation of the Sobol sequence initialized in the first step. This approach is analogous to implicit averaging, advocated as one of the optimization strategies in noisy settings within the field of evolutionary optimization [23]. It assumes that gathering the values of the distance function in the neighborhood of previous parameter sets with a good fit can implicitly compensate for the noisiness of the function's values. This enables us to search a broader range of parameter values and save model calls. Alternatively, averaging out the noise could be achieved by calculating the distance metrics for a single

[2] Search ranges for all parameters need to be scaled in the same way first.

DOI: 10.1007/978-3-030-77967-2_27

parameter set using different realizations of the noise term. We consider this alternative inefficient, which is supported by the results in Section 3.2.

Steps 2-6 can be repeated until the computational budget is fully exhausted. After the budget is finished, from the most promising neighborhoods we pick the centroid with the lowest value of the distance function $\mathcal{D}$ as the final set of calibrated parameters $\hat{\theta}$. Importantly, this is not the point that minimizes $\mathcal{D}$. Our calibration process may have uncovered parameter sets with a lower distance metric. However, the chosen parameter set offers a low value of $\mathcal{D}$ within a neighborhood of other low values of the distance function. We find that concentration purely on minimizing $\mathcal{D}$ may be counterproductive, as it does not necessarily lead to a lower value of the fitness function $\mathcal{F}$. Simulations in Section 3.2 provide an illustration of that issue.

The full proposed procedure is summarized as Algorithm 1. Algorithm parameters include:

- n: number of available model calls per 1 iteration of the algorithm,
- β: number of the parameter sets corresponding with the lowest loss function values to be left for further processing in each iteration,
- α: number of nearest neighbors used to create local neighborhoods,
- γ: number of the most promising neighborhoods to be allocated additional model calls.

Algorithm 1 Filtered Neighborhoods optimization algorithm

```
Result: calibrated parameter set $\hat{\theta}$
draw the initial sample of $n$ HAM parameter sets;
repeat
    leave only $\beta$ parameter sets with the lowest values of the distance metric;
    create $\beta$ local neighborhoods, taking $\alpha$ nearest neighbors for each
      parameter set in the sample;
    use the acquisition function to rank the neighborhoods;
    if performed model calls $+n >=$ computational budget then
        set the output parameter vector to the values associated with the
          lowest value of the loss function;
        break
    end
    distribute the next-round budget of $n$ simulations among $\gamma$ most
      promising neighborhoods;
    draw the next sample of $n$ HAM parameter sets within the chosen
      neighborhood areas;
until break;
```

From a functional perspective, parameter n regulates the computational cost of the optimization process. β governs filtering out the least promising parameter sets, with lower values meaning stricter filtering. α and γ are directly related

DOI: 10.1007/978-3-030-77967-2_27

to the exploration-exploitation trade-off - a crucial issue in optimization [5]. Higher α increases the size of the neighborhoods, thus promoting exploration over exploitation. Similarly, higher γ spreads the available model calls over more neighborhoods, also promoting exploration.

3 Calibration results of a benchmark agent-based model

3.1 The Brock and Hommes 1998 model

The Brock and Hommes 1998 model (abbreviated further as BH98) is one of the most important benchmark financial ABMs [11]. It served as a platform for testing calibration methods in the recent years [18, 22]. This HAM models the interaction of traders following h strategies, each having the following forecast $f_{h,t}$ for the market's price deviation x_t from the fundamental price p^*:

$$f_{h,t} = g_h x_{t-1} + b_h \tag{2}$$

g_h constitutes a trend parameter and b_h is a bias parameter. Market price dynamics within this model is characterized by the following 3 equations:

$$(1+r)x_t = \sum_{h=1}^{H} n_{h,t}(g_h x_{t-1} + b_h) + \epsilon_t \tag{3}$$

$$n_{h,t} = \frac{\exp(\beta U_{h,t-1})}{\sum_{h=1}^{H} \exp(\beta U_{h,t-1})} \tag{4}$$

$$U_{h,t-1} = (x_{t-1} - Rx_{t-2}) \left(\frac{g_h x_{t-3} + b_h - Rx_{t-2}}{a\sigma^2} \right) \tag{5}$$

Symbol r denotes the risk-free rate, with $R = 1 + r$. $U_{h,t-1}$ is the fitness measure of strategy h assessed at the beginning of period t. The noise term ϵ_t depicts the uncertainty about the economic fundamentals. a is the risk aversion parameter and β is the intensity of choice parameter. σ^2 denotes the variance of excess return. $n_{h,t}$ is the fraction of market participants following strategy h.

Figure 2 depicts the noisiness of BH98, presenting two time series output by the model with the same parameter values, but different random seeds.

3.2 Calibration process and results

The recent comparison of calibration methods by Platt [22] becomes our benchmark, as it tests multiple ABM calibration methods (MSM among them) on a few ABMs, including BH98. In the calibration process we focus on analyzing the performance of our optimization algorithm, relying on Platt for other choices necessary in the calibration process.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_27

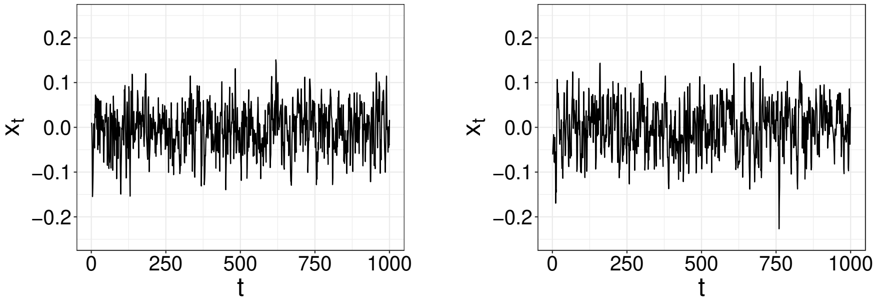

Fig. 2. Brock and Hommes 1998 model output with the same parameter values and different random seeds.

We follow the most complex case involving the BH98 model presented in [22], choosing:

- 4 strategies interacting in the model;
- 4 out of 11 model parameters to be calibrated;
- 7 moments used in calibration: the variance of the raw series, kurtosis of the raw series, autocorrelation coefficients of the raw series, absolute value series and squared series at lag 1, and the autocorrelation coefficients of the absolute value series and squared series at lag 5;

As pointed out by [6], calibrating only a subset of parameters is common practice due to the size of the search space growing exponentially with additional parameters. The chosen moments are supposed to constitute a measure of stylized facts observable in financial time series [21].

The distance function $\mathcal{D}$ used to evaluate the distance between the set of simulated moments and the moments for the "true" dataset is chosen to be the weighted sum of squared errors. Following [22], the weighting matrix is the inverse of the covariance matrix of the "true" moments (Newey-West estimator).

The final calibrated set of parameters obtained through the optimization algorithm is compared with the "true" parameter set using the following fitness function:

$$\mathcal{F}\left(\theta,\hat{\theta}\right) = ||\theta - \hat{\theta}||_2 \tag{6}$$

The experimental process assumes first generating 100 time series from BH98, 1,000 observations each. They are generated using a single set of model parameters and 100 different seeds for the random number generator. These will be treated as the "true" data and the optimization algorithm will be used to uncover the values of model parameters used when generating the data. Using 100 "true" datasets, differing just because of the random noise terms and not the model parameters, enables us to calculate confidence intervals for various algorithm performance statistics. This is a key difference between the experimental procedure used in this paper and [22], who uses only a single run of the BH98 as his "true" dataset.

We opt for a rather modest computational budget, assuming 5,500 to 6,000 model calls per 1 ”true” dataset: 1 initialization round and 10 optimization rounds with 500 calls per round. Some algorithm setups use 1,000 model calls in the initialization round only. As in [22], each model call generates a simulated run of 1,000 observations, for which the 7 aforementioned moments are calculated and compared with the moments of a ”true” dataset. In accordance with the suggestions of the algorithm, each call uses a different model parameter set and a different realization of the noise term. This differs from Platt [22], who generally uses 10 optimization rounds with 1,000 different parameter sets (compared to 500 in this paper), but also performs 250 Monte Carlo trials per each parameter set to average out the random noise. His technique drives the total budget to 2,500,000 model calls.

The ”true” parameter set used in the calibration process follows [22] and is depicted in Table 1. Out of these parameters, the calibration process seeks the values of g_2, b_2, g_3 and b_3, treating the other parameters as known.

Table 1. True model parameters.

Parameter	g_1	b_1	g_2	b_2	g_3	b_3	g_4	b_4	r	β	σ
Value	0	0	0.6	0.2	0.7	-0.2	1.01	0	0.01	10	0.04

Within the experiment, 10 sets of algorithm parameters are tested, keeping the computational budget stable with $n = 500$. Some algorithm setups utilize 1,000 instead of 500 model calls in the initial round only. The acquisition function employed in the algorithm is the well-known Probability of Improvement [3]. The parameter space is shown in Table 2.

Table 2. Filtered Neighborhoods algorithm parameter sets used in the calibration process.

Parameter	n	β	α	γ
Set 1	500	250	30	20
Set 2	500	250	10	20
Set 3	500	250	50	20
Set 4	500	100	30	20
Set 5	500	500	30	20
Set 6	500	250	30	10
Set 7	500	250	30	50
Set 8	500 (1,000 for initial draw)	250	30	20
Set 9	500 (1,000 for initial draw)	200	50	50
Set 10	500 (1,000 for initial draw)	100	50	50

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_27

Competing HAM calibration methods include:

- Method of Simulated Moments (MSM),
- Generalized Subtracted L-divergence, which measures the distance between distributions of patterns in the ”true” and simulated time series [19] (GSL),
- Markovian Information Criterion (MIC), based on the Kullback-Leibler distance between the ”true” and simulated data [1],
- Bayesian estimation approach presented by [12] (BE).

MSM, GSL and MIC are all optimized using either Particle Swarm Optimization (MSM/PS, GSL/PS, MIC/PS) or the approach of Knysh and Korkolis [17] (MSM/KK, GSL/KK, MIC/KK). Additionally, the newly proposed Filtered Neighborhoods algorithm is used within the Method of Simulated Moments (MSM/FN).

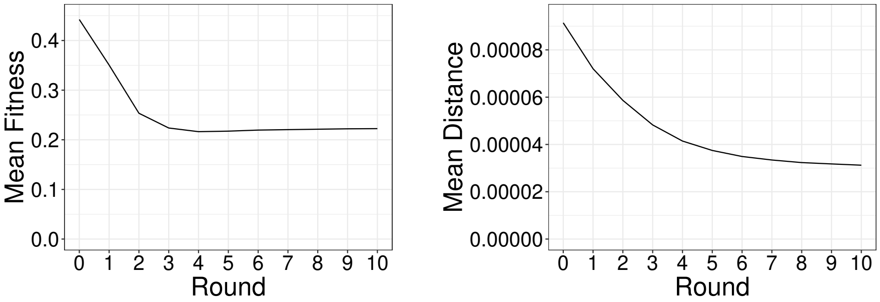

Fig. 3. Comparison of the mean fitness function $\mathcal{F}$ value and mean distance function value $\mathcal{D}$ per subsequent optimization round for all tested Filtered Neighborhoods parameter sets.

Figure 3 shows the mean fitness function $\mathcal{F}$ value for all 10 tested Filtered Neighborhoods parameter sets in each optimization round. Fitness for MSM/FN seems to converge already in round 4 at the value of 0.22. This is less than half of the error value achieved by the direct competitors, MSM/PS (0.45) and MSM/KK (0.54). What is more, instead of using 2,500,000 model calls, MSM/FN uses ca. 5,000-5,500 calls for 10 rounds - and only 2,000-2,500 calls if stopped after 4 rounds. Interestingly, continuing the optimization beyond round 4 does decrease the value of the distance function $\mathcal{D}$, but does not improve HAM parameter calibration accuracy.

The best discovered MSM/FN parameter set is Set 10 in Table 2. It is characterized by the increased number of model calls in the initial optimization round (1,000), prominent filtering of the dataset ($\beta = 100$), and leaning towards exploration ($\alpha = 50$, $\gamma = 50$). The average achieved fitness function value across 100 different realizations of the ”true” time series was 0.172, with the 99% confidence interval being 0.152-0.193. This is a significantly better result than obtained by

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_27

any of the competing methods, with the exception of Bayesian Estimation. However, it is not significantly worse than the performance of Bayesian Estimation, notwithstanding the lack of confidence intervals provided for the results presented in [22].

Table 3. Calibration results.

Method	g_2	b_2	g_3	b_3	$\mathcal{F}$
True θ	0.6	0.2	0.7	-0.2	-
MSM/FN	0.6096	0.1609	0.7037	-0.1598	**0.0570**
MSM/KK	0.8742	0.2424	0.2424	-0.2040	0.5352
MSM/PS	1	0.1941	0.5015	-0.2174	0.4469
GSL-div/PS	0.6047	0	0.5277	0	0.3312
GSL-div/KK	0.5575	0	0.5529	0	0.3216
MIC/KK	0.3663	0.2431	0.8766	-0.2387	0.2985
MIC/PS	0.8105	0.2278	0.7430	-0.2186	0.2173
BE	0.4651	0.2468	0.6251	-0.2388	0.1658

What is more, mean calibrated HAM parameter values for MSM/FN are very close to the ”true” ones and compare favorably with the competition - as evident in Table 3. Fitness value for the mean parameters calibrated with MSM/FN is only 0.057. Due to the noisiness of the HAM model, in 12 out of 100 cases the value of the distance function $\mathcal{D}$ for the calibrated parameter set turns out to be lower than for the ”true” parameter set.

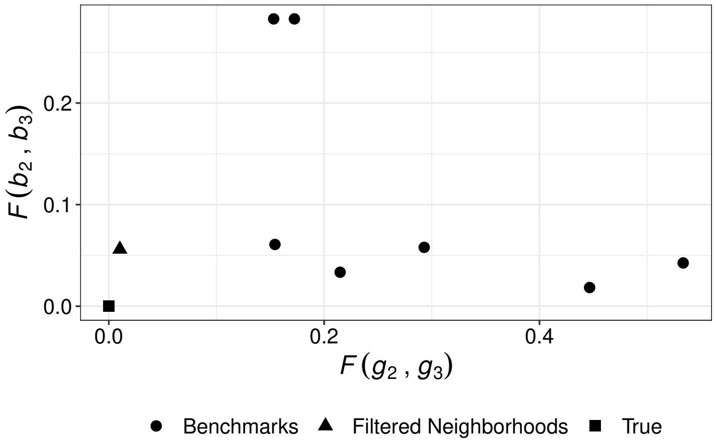

Fig. 4. Fitness metrics for the pairs of calibrated b and g parameters.

DOI: 10.1007/978-3-030-77967-2_27

Figure 4 presents the fitness value of the examined models decomposed into two fitness metrics for both pairs of calibrated b and g parameters. On average, the Filtered Neighborhoods algorithm finds both g parameters with almost perfect accuracy. Most of the algorithm error comes from missing the ”true” value of b parameters.

4 Conclusions and further research directions

The newly proposed Filtered Neighborhoods optimization method for the Method of Simulated Moments clearly outperforms both Particle Swarm Optimization and the approach of Knysh and Korkolis within the context of the BH98 model. The quality of the calibration is superior to most of the other benchmarks as well, and similar to that of Bayesian Estimation. All that is achieved while using roughly 500 times less model calls, which saves a considerable amount of computational time.

The advantage of the new method holds for a range of different algorithm parameter values. Stronger filtering and pursuing exploration seem to be beneficial for calibration accuracy.

In this paper, the efficiency of our method has been proved only for the BH98 model and for a specific calibration issue involving 4 parameters. Still, as postulated in [21], the best choice of a calibration framework for an agent-based model might be model-specific. Our method seems to be well-suited to the problem at hand, and the conclusions seem rather robust to MSM/FN parameter changes. However, more work is needed to examine the optimal choice of the algorithm’s parameter values. Other acquisition functions (including function portfolios) should also be tested. A much larger effort altogether (and considerable computation costs) would be necessary to prove the Filtered Neighborhoods algorithm’s reliability over a broader range of parameter sets of BH98, as well as other important HAMs - or any other noisy optimization problems. The results for BH98 definitely make this avenue worth pursuing.

On a more general note, the presented calibration results indicate how a long pursuit of the minimum of the observable distance function may not contribute to finding the minimum of the unobservable fitness function - the true purpose of the whole calibration process. The relationship between the distance metric and the fitness metric should be deeply examined and taken into account in the optimization process. This may not only improve calibration accuracy, but lessen the necessary computational workload.

References

1. Barde, S.: A practical, accurate, information criterion for nth order Markov processes. Computational Economics **50**, 281–324 (2017). https://doi.org/10.1007/s10614-016-9617-9
2. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research **13**, 281–305 (2012)

3. Brochu, E., Cora, V.M., de Freitas, N.: A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. CoRR **abs/1012.2599** (2010)
4. Brock, W.A., Hommes, C.H.: Heterogeneous beliefs and routes to chaos in a simple asset pricing model. Journal of Economic Dynamics & Control **22**, 1235–1274 (1998). https://doi.org/10.1016/S0165-1889(98)00011-6
5. Chen, J., Xin, B., Peng, Z., Dou, L., Zhang, J.: Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans **39**(3), 680–691 (2009). https://doi.org/10.1109/TSMCA.2009.2012436
6. Chen, S.H., Chang, C.L., Du, Y.R.: Agent-based economic models and econometrics. Knowledge Engineering Review **27**(2), 187–219 (2012). https://doi.org/10.1017/S0269888912000136
7. Chen, Z., Lux, T.: Estimation of sentiment effects in financial markets: a simulated method of moments approach. Computational Economics **52**, 711–744 (2018). https://doi.org/10.1007/s10614-016-9638-4
8. Dieci, R., He, X.Z.: Heterogeneous agent models in finance. In: Hommes, C., LeBaron, B. (eds.) Handbook of computational economics, vol. 4, chap. 5, pp. 257–328. Handbooks in economics, Elsevier (2018). https://doi.org/10.1016/bs.hescom.2018.03.002
9. Dutang, C., Savicky, P.: randtoolbox: Generating and Testing Random Numbers (2020), r package version 1.30.1
10. Ellen, S., Verschoor, W.: Heterogeneous beliefs and asset price dynamics: a survey of recent evidence. In: Jawadi, F. (ed.) Uncertainty, expectations and asset price dynamics: essays in honor of Georges Prat, Dynamic modeling and econometrics in economics and finance, vol. 24, pp. 53–79. Springer (2018). https://doi.org/10.1007/978-3-319-98714-9_3
11. Fagiolo, G., Guerini, M., Lamperti, F., Moneta, A., Roventini, A.: Validation of agent-based models in economics and finance. In: Beisbart, C., Saam, N. (eds.) Computer simulation validation. Simulation foundations, methods and applications, chap. 31, pp. 763–787. Springer (2019). https://doi.org/10.1007/978-3-319-70766-2_31
12. Grazzini, J., Richiardi, M.G., Tsionas, M.: Bayesian estimation of agent-based models. Journal of Economic Dynamics and Control **77**, 26–47 (2017). https://doi.org/10.1016/j.jedc.2017.01.014
13. He, X.Z.: Recent developments in asset pricing with heterogeneous beliefs and adaptive behaviour of financial markets. In: Bischi, G., Chiarella, C., Sushko, I. (eds.) Global analysis of dynamic models in economics and finance: essays in honour of Laura Gardini, pp. 3–34. Springer, Berlin Heidelberg (2013). https://doi.org/10.1007/978-3-642-29503-4_1
14. Hommes, C.H.: Heterogeneous agent models in economics and finance. In: Tesfatsion, L., Judd, K.L. (eds.) Handbook of computational economics, vol. 2, chap. 23, pp. 1109–1186. Handbooks in economics, Elsevier (2006). https://doi.org/10.1016/S1574-0021(05)02023-X
15. Hong, H.S., Hickernell, F.J.: Algorithm 823: Implementing scrambled digital sequences. ACM Transactions on Mathematical Software **29**(2), 95–109 (2003). https://doi.org/10.1145/779359.779360
16. Klein, A., Falkner, S., Bartels, S., Hennig, P., Hutter, F.: Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets. In: Singh, A., Zhu, J. (eds.) Proceedings of the 20th International Conference on Artificial Intelligence

and Statistics. Proceedings of Machine Learning Research, vol. 54, pp. 528–536. PMLR, Fort Lauderdale, FL, USA (2017)

17. Knysh, P., Korkolis, Y.: Blackbox: A procedure for parallel optimization of expensive black-box functions. CoRR **abs/1605.00998** (2016)
18. Kukacka, J., Barunik, J.: Estimation of financial agent-based models with simulated maximum likelihood. Journal of Economic Dynamics and Control **85**, 21–45 (2017). https://doi.org/10.1016/j.jedc.2017.09.006
19. Lamperti, F.: An information theoretic criterion for empirical validation of simulation models. Econometrics and Statistics **5**, 83–106 (2018). https://doi.org/10.1016/j.ecosta.2017.01.006
20. LeBaron, B.: Agent-based computational finance. In: Tesfatsion, L., Judd, K.L. (eds.) Handbook of computational economics, vol. 2, chap. 24, pp. 1187–1233. Handbooks in economics, Elsevier (2006). https://doi.org/10.1016/S1574-0021(05)02024-1
21. Lux, T., Zwinkels, R.C.J.: Empirical validation of agent-based models. In: Hommes, C., LeBaron, B. (eds.) Handbook of computational economics, Handbooks in economics, vol. 4, chap. 8, pp. 437–488. Elsevier (2018). https://doi.org/10.1016/bs.hescom.2018.02.003
22. Platt, D.: A comparison of economic agent-based model calibration methods. Journal of Economics Dynamics and Control **113**, 103859 (2020). https://doi.org/10.1016/j.jedc.2020.103859
23. Rakshit, P., Konar, A., Das, S.: Swarm and evolutionary computation. Journal of the American Statistical Association **33**, 18–45 (2017). https://doi.org/10.1016/j.swevo.2016.09.002
24. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N.: Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE **104**(1), 148–175 (2016). https://doi.org/10.1109/JPROC.2015.2494218
25. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 25, pp. 2951–2959. Curran Associates, Inc. (2012)