

RESEARCH ARTICLE

# Beyond Corroboration: Strengthening Model Validation by Looking for Unexpected Patterns

Guillaume Chérel<sup>1,2#a\*</sup>, Clémentine Cottineau<sup>1,3#b</sup>, Romain Reuillon<sup>1,2#a</sup>

**1** Géographie-Cités, CNRS, Paris, France, **2** ISC-PIF, Paris, France, **3** Centre for Advanced Spatial Analysis, UCL, London, United Kingdom

#a Current address: Institut des Systèmes Complexes Paris Île-de-France, 113 rue Nationale, 75013, Paris, France

#b Current address: Géographie-Cités, 13 rue du four, 75006, Paris, France

\* [guillaume.cherel@iscpif.fr](mailto:guillaume.cherel@iscpif.fr)



CrossMark  
click for updates

**OPEN ACCESS**

**Citation:** Chérel G, Cottineau C, Reuillon R (2015) Beyond Corroboration: Strengthening Model Validation by Looking for Unexpected Patterns. PLoS ONE 10(9): e0138212. doi:10.1371/journal.pone.0138212

**Editor:** Frederic Amblard, Université Toulouse 1 Capitole, FRANCE

**Received:** April 10, 2015

**Accepted:** August 27, 2015

**Published:** September 14, 2015

**Copyright:** © 2015 Chérel et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** Source code for replicating the experiments is available from the github repository <https://github.com/guillaumecherel/PSEExperiments>.

**Funding:** The study was funded by the ERC advanced grant 269826 GeoDiverCity (PI Denise Pumain).

**Competing Interests:** The authors have declared that no competing interests exist.

## Abstract

Models of emergent phenomena are designed to provide an explanation to global-scale phenomena from local-scale processes. Model validation is commonly done by verifying that the model is able to reproduce the patterns to be explained. We argue that robust validation must not only be based on corroboration, but also on attempting to falsify the model, i.e. making sure that the model behaves soundly for any reasonable input and parameter values. We propose an open-ended evolutionary method based on Novelty Search to look for the diverse patterns a model can produce. The Pattern Space Exploration method was tested on a model of collective motion and compared to three common a priori sampling experiment designs. The method successfully discovered all known qualitatively different kinds of collective motion, and performed much better than the a priori sampling methods. The method was then applied to a case study of city system dynamics to explore the model's predicted values of city hierarchisation and population growth. This case study showed that the method can provide insights on potential predictive scenarios as well as falsifiers of the model when the simulated dynamics are highly unrealistic.

## Introduction

Writing models to explain emergent phenomena is trying to figure out what mechanisms at the local scale create the patterns observed at the global scale. Such models, often computer-based and requiring simulation, are keys to the scientific inquiry as soon as the system's description involves a local and a global scale and it is not trivial to explain the global patterns from the local dynamics. They are increasingly developed in a wide range of disciplines from physics to biology, ecology, and social sciences. A long list of examples includes species evolution [1], problem solving in social insects [2], the dynamics of neural networks [3], social segregation [4], population dynamics [5], collective motion [6], and many more.

A classical model validation approach is to write a model of the local scale dynamics and interactions, calibrate it, run simulations, and compare the outputs to the data. These steps lead us to conclude whether the proposed model can explain the patterns observed in nature or not, and revise it as necessary [7–10]. But to say that the model can reproduce some patterns does not entail that it accurately reflects how they are produced in nature or society [11, 12]. A given global pattern can sometimes be explained by different models involving different local dynamics. An approach to validation based on mere corroboration is insufficient to build robust explanatory and predictive models. We need more rigorous practices. In this paper, we propose a method contributing to this need for simulated models by searching for unexpected model behaviours.

How can we test the validity of a model? We can make a useful detour by Karl Popper's falsifiability principle. A theory yields specific statements which it allows or forbids. Allowed specific statements, when confirmed by empirical validation, corroborate the theory. Forbidden statements falsify it. Popper says that accumulating observations which corroborate the theory does not help in testing its validity. Consider the general statement "All swans are white". Observing any number of white swans never discards the possibility of observing a black swan in the future. A single observation of a forbidden statement, on the contrary, immediately disproves it. It follows that one can only decide between competing theories by looking for their potential falsifiers.

Modelling emergent phenomena can be viewed as a specific case of the conceptual framework which Popper uses for his argument (Fig 1). A model is a general statement about how the world works. The model yields specific statements as individual simulations, i.e. associations of input and output values. The individual simulations can be partitioned into two sets: those which correspond to empirical observation, and those which contradict it. Members of the former corroborate the model, while members of the latter falsify it. It is of course important to ensure that a model is able to reproduce the patterns it is designed to explain. And it is equally important, following Popper's argument, to look for simulations that may have the power to falsify the model, those which display an association of input values and output patterns which contradict data. We use the word *pattern* to denote a description of a simulation run. For example a simulation of a spatially explicit population dynamics model can be described in terms of population growth and spatial aggregation. All distinct combinations of values for these two variables are as many distinct patterns.

Models with high complexity can produce patterns that are unexpected even to the eye of the modeller. An unexpected pattern can either be a falsifier of the model if it is contradicted by subsequent empirical observation, or a prediction of the model if it is validated by it. To search for the unexpected patterns a model can produce in simulation is to search for both at the same time.

In the context of model building, searching for unexpected patterns should be thought of as one step in a broader iterative process. In this process, a model is designed, the pattern space explored, unexpected patterns (in particular falsifiers) used to revise the model assumptions, and we start over with exploring the pattern space with the revised model. The loop goes on until we get to a satisfying model.

The problem of searching for unexpected patterns is apart from the problem of calibration where some particular output is expected, since we do not know explicitly what to look for. We propose to approach it by looking more generally for the diverse patterns that our models can produce. The pattern obtained in a simulation depends on the model inputs (e.g. initial state) and the parameter values (controlling the model dynamics). Exploring the inputs can be interesting when studying what patterns can be produced from different initial conditions. Exploring the parameters is needed when there is uncertainty about their calibration (i.e. insufficient

Theory (All swans are white.)	Specific statements	Allowed (I see white swans.)
		Forbidden (I see a black swan.)
Model	Simulations (input and output values)	Coherent with empirical observation
		In contradiction with empirical observation

**Fig 1. Modelling and simulation share a conceptual structure with Popper’s falsifiability argument.**

doi:10.1371/journal.pone.0138212.g001

data). Input and parameter variables form together what we call here for simplicity the parameter space of the model (without loss of generality, inputs can be seen as model parameters, or equivalently variable parameters can be seen as model inputs). Thus while the exploration is driven within the pattern space in search for diversity, it is fundamentally a problem of parameter space exploration: how to explore the parameter space to find the various patterns, including those that are unexpected?

Such an exploration is no trivial task due to the common difficulties with computer-based models of complex systems: the high dimensionality of the parameter space, non-linearity between inputs and outputs, and stochasticity. High dimensionality (tens or hundreds of parameters) makes the parameter space prohibitively big to try out all possible parameter values (or a regular sampling in the case of real valued parameters). Non-linearity entails that interesting output values may correspond to very small regions of the parameter space, making them difficult to look for. Stochasticity makes it necessary to replicate the simulations. For models which take several seconds to run in simulation, a clever exploration method to reduce the number of simulations is necessary.

The active research field of experiment design aims at solving the problem of sampling vast parameter spaces. Some methods try to tackle the problem of dimensionality of the parameter space with a priori samplings, such as the well known Latin hypercube sampling or LHS (see examples of sampling methods and comparisons in [13–15]). Others try to solve it in an adaptive way, by choosing iteratively where to sample next based on information collected previously [16–18]. All start with the base problem of sampling efficiently (with a number of points that is tractable in simulation) the parameter space in order to get information about the relationships between the control parameters and the observables. They solve the problem by not exploring the parameter space exhaustively, depending on the information they do not need, given their particular objectives.

Our problem focuses on exploring the pattern space in search for unexpected patterns. We do not have any constraint about the parameter space. This transfers the problem of dimensionality from the parameter space to the pattern space. While it may be hard to reduce the number of free parameters of a model of complex system, modellers are free to choose what variables to measure on a simulation, i.e. how many dimensions the pattern space has. Typically, a 2 or 3 dimensional space has both the advantage of being tractable and easily visualised.

Evolutionary methods are well suited for this setting because they can perform a search through the parameter space based on information gathered in the pattern space by simulation, and do so without any a priori knowledge of the relationship between the parameter space and the pattern space. They were successfully used for model calibration when data about some parameters is lacking but there is an idea of global patterns to be expected [19–22]. Evolutionary calibration methods are all based on a priori ideas of the patterns one would like to obtain. The a priori target pattern can be based on data to reproduce an observed scenario. It can also be designed to represent an expected behaviour and used to assess the model's ability to reproduce it (such as the V formation of flocks in [22]). An objective function is designed in terms of a distance between a model run and the target pattern, and evolutionary methods are used to tune the parameters to minimise it. Our problem is different in that, rather than targeting a particular pattern, we want to find the various ones a model can produce.

Miller [23] proposed a model breaking method using an evolutionary algorithms which looks for how far a model behaviour can depart from a calibrated behaviour when the parameters are kept within the vicinity of their calibrated values. Stonedahl and Wilensky used a similar approach for the purpose of sensitivity analysis [24]. This idea is getting closer to our objective of finding unexpected patterns, but has three limitations regarding this goal. First, it restricts the search to a small region of the search space. Second, it relies on a target pattern to which other produced patterns are compared. Third, like the calibration methods, it uses an objective-based search to maximise the distance from the target pattern. The objective-based search by nature progressively narrows the exploration towards a smaller region of the search space. While looking for unexpected patterns, we would rather not limit ourselves to just finding one, but instead look for all the different unexpected patterns we can find in the course of a single exploration. Rather than narrowing the search as objective-based methods do, we propose to continually widen the exploration in search for the diverse patterns a model can produce.

We propose to solve this problem with a new open-ended evolutionary algorithm based on Novelty Search [25, 26]. Novelty Search does not rely on an objective function. It generates successive populations of solutions to a problem by breeding the most novel individuals of the current population. The novelty of current individuals is measured with respect to the current population and an archive of past individuals, and reflects the novelty of patterns (not parameter values). Novelty Search continually drives the exploration of the parameter space towards areas which produce novel patterns. The method we propose here, called Pattern Space Exploration (PSE), is novelty search adapted to the exploration of agent-based and more generally computer-based models of complex systems. In standard Novelty Search [26], the novelty measure is based on the average distance of an individual to its  $k$  nearest neighbours in the pattern space. In PSE, novelty relies on a hit-map counter. The hit-map is a discretisation of the pattern space into regular cells, and we count the number of individuals which belong to each cell throughout the exploration. The novelty measure of an individual is based on the inverse value of its corresponding cell count. The less a cell has received individuals before, the more novel is an individual standing on it. This variation offers several advantages for our purpose. First, defining a distance measure in a multi-dimensional space is not a trivial task, in particular when the different dimensions represent unrelated variables. The hit-map approach alleviates this problem by suppressing the distance measure, and relying instead on the discretisation of the pattern space which is done independently for each dimension. Second, computing the  $k$  nearest neighbours is computationally costly (for example, finding the nearest neighbour in a KD-tree has complexity  $O(\log(N))$ , where  $N$  is the number of individuals), and constitute a serious bottleneck in an evolutionary algorithm because it cannot be parallelised (while the individuals evaluations are independent and can be). The only operations needed with the hit-

map are accessing or incrementing the counter at a given cell, and are done in constant time. Third, the hit-map naturally offers a measure of the exploration progress, the *volume discovered* in the pattern space, approximated by counting the number of cells which counters are positive.

In this paper we present PSE and two sets of experiments. The first one was designed to test the method's efficiency in exploring the pattern space of a model, i.e. in finding a wide range of different patterns. We used a model of collective motion and compared PSE to common a priori sampling methods (LHS, regular grid, Sobol). Collective motion is a well studied phenomenon in physics and the different patterns that can be exhibited is well described. The second experiment was a real modelling scenario with a recently developed model of system of cities—MARIUS (Model of Agglomerations within Imperial Russia and the (former) Soviet Union) [27]. We used PSE to explore thematic questions regarding the possible outcomes that the model is able to simulate. This experiment illustrates one iteration of the iterative modelling process proposed above (the pattern space exploration and the analysis of unexpected patterns to revise the model's assumptions). The following sections present the algorithm and the two experiments along with their results.

### Algorithm description

The Pattern Space Exploration method (PSE) is an open-ended evolutionary algorithm. It is composed of the common building blocks of evolutionary algorithms (see algorithm 1): a population of individuals whose genomes each encode a particular element of the search space, a selection method to select individuals from which a new population will be created, a crossover and mutation mechanisms to create a new individual from previously selected ones, and an elitism mechanism to filter all the individuals (newly created and the old population) to be kept and form the new population. As the algorithm progresses, unwanted individuals are discarded and forgotten as new ones are added to the population. An archive is used to store information about past populations. Each of these building blocks can be specified in different manners in order to build different evolutionary algorithms. Below, we will describe how we designed each building block to implement the PSE method.

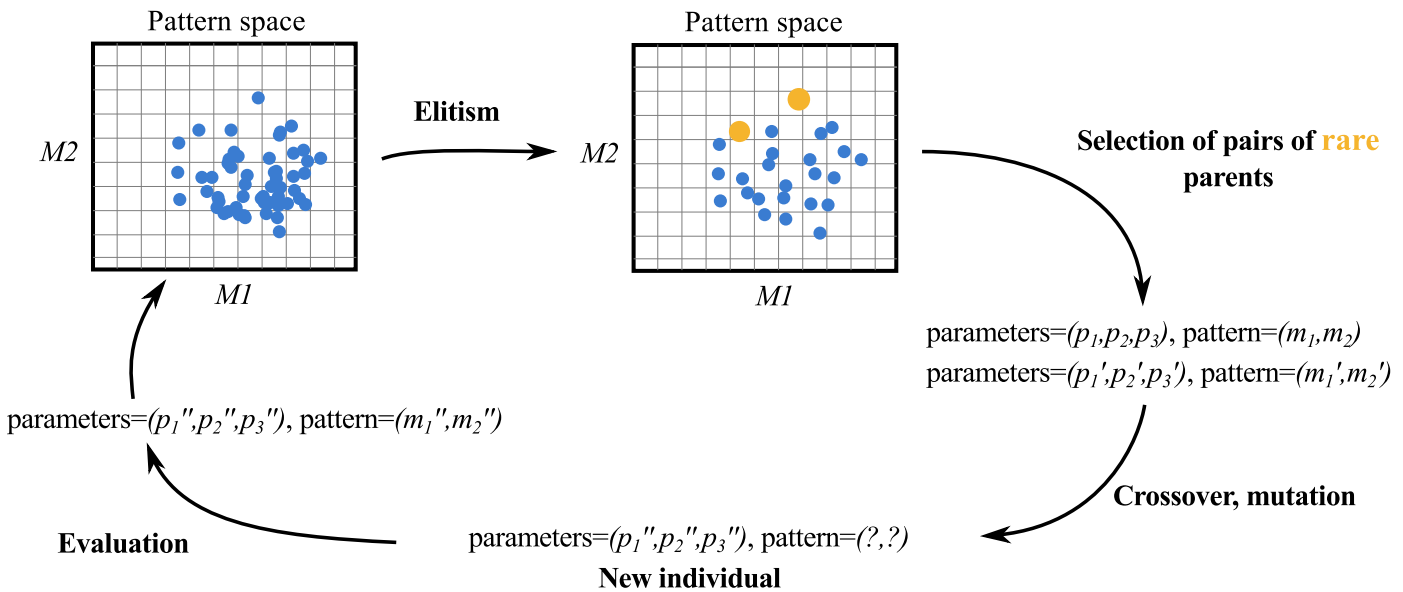
```

1 Generate  $\mu$  genotypes by sampling the genotypic space randomly
2 Evaluate them
3 while stop condition not reached do
4   for  $i \leftarrow 1$  to  $\lambda$ ,  $s$  the number of new individuals to create do
selection 5     Select 2 individuals from the population
crossover 6     Create a new genotype by crossover from the 2 selected
              individuals' genotypes
mutation 7     Mutate the new genotype
8   end
evaluation 9   Compute the offsprings' phenotypes
elitism 10   Filter the individual of the whole population (parents and
              offsprings) based on their phenotype
11 end

```

**Algorithm 1:** General scheme of an evolutionary algorithm

PSE is illustrated in Fig 2. The algorithm starts by generating individuals with random genomes representing parameter values (the *new individual* stage in the figure). For each, a simulation is run and the pattern is measured (evaluation). This results in as many patterns as generated individuals. At the elitism stage, individuals are filtered to keep only those that are significantly different from one another with respect to their patterns. Pairs of individuals are



**Fig 2. General scheme of the Pattern Space Exploration method.** M1 and M2 represent the dimensions of the pattern space. The  $p_i$  represent individual parameter variables, and the  $m_i$  represent pattern variables. The pattern space is discretised into cells and rare (novel) parents are selected based on the hit count of their corresponding cell. New individuals are generated by crossover and mutation and evaluated to get their pattern. Newly evaluated individuals increment the hit count of their respective cell. All individuals (parents and offsprings) are filtered at the elitism stage so only one per cell remains, randomly picked for each cell.

doi:10.1371/journal.pone.0138212.g002

selected based on how rare their patterns are relative to the current population and past ones (rarity is PSE's equivalent of the novelty measure). Each element's genome from each pair is recombined to form a new genome, which is then further mutated. A new generation of individuals is thus created, and the loop continues. The main idea is that by selecting parents whose patterns were rarely observed previously, we increase the chances to find patterns yet undiscovered.

### Individuals (genotype, evaluation, phenotype)

An individual is described by a genome and a phenotype. Its genome encodes an element of the search space. The phenotype of the individual is computed given its genotype through evaluation. The phenotype thus reflects the behaviour of the individual.

In PSE, a genotype encodes a value for each parameter. An evaluation corresponds to a simulation run with the parameter values taken from the genome. A phenotype encodes a pattern measured on the simulation run.

Here, both the genotypic and phenotypic spaces are multidimensional real spaces, respectively  $\mathbb{R}^K$  and  $\mathbb{R}^M$ , where  $K$  is the number of parameters of a model, and  $M$  is the number of descriptive variables of a simulation. The restriction to real values is done for simplicity. The experiments below seamlessly use integers in some of the parameters dimensions by casting them to doubles to construct the genotypes, without any alteration of the algorithm. The use of categorical (non-ordinal) or more complex parameter types would require using other more appropriate random sampling methods (for population initialisation), as well as crossover and mutation operators than those used here. In the pattern space, the algorithm can also deal with integers and categorical values by simply mapping them to real values. More complex types might pose scalability issues if they implicitly add dimensions to the pattern space, and might be difficult to discretise (see archive below).

When the model dynamics or initialisation are stochastic, it is necessary to replicate each simulation for statistical robustness. In this case, the median over the replications is used for each of the  $M$  phenotypic variable.

## Initialisation

The first generation of  $\mu$  individuals is created with random genomes, by drawing uniformly a value in each of the model parameter domain.

## Archive

PSE uses a hit map archive to keep a trace of how much the different regions of the phenotypic space have been explored. This information is used at the selection stage to select rare individuals, and at the elitism stage to avoid the accumulation of too many individuals in the same region of the space (see below). The archive consists in a user-specified discretisation of the phenotypic space along each dimension, thus forming a set of  $M$ -dimensional cells covering the phenotypic space. The discretisation is provided as a step size for each phenotypic dimension. The resulting discretisation in each domain is “anchored” on 0 (a step size of 1 places the discrete bounds at . . . -2, -1, 0, 1, 2 . . .). Each phenotype can be naturally located in its corresponding cell. Each cell has an associated hit count summing the number of individuals’ phenotypes which were located in the cell throughout the evolutionary process. The hit map is updated between the evaluation stage and the elitism stage. The hit map is encoded as a sparse dictionary where keys represent cell positions and the values represent the hit counts. Keys are added to the hit map when new phenotypes are found. Any key not present in the hit map implicitly has a value 0. This allows to deal with the discretisation of an infinite space without having to set min and max bounds. Bounds can be set artificially to focus the exploration on a certain region, as explained in the experiments below.

## Selection

Traditionally, classical objective-based evolutionary algorithms rely on a measure of fitness to select the individuals. The fitness reflects the quality of an individual, that is how well it performs regarding the objective. The selection stage then consists in choosing fit individuals in order to move the search towards good solutions. Here however, our objective is to find diverse phenotypes. There is no notion of fitness. In PSE, we select the individuals whose phenotypes are rare. By selecting these, the exploration will focus on the areas of the phenotypic space that were yet little explored. The rarity of a phenotype is given by the inverse of the hit count of the corresponding cell in the hit map archive.

Selection is performed as a binary tournament. Two individuals are picked at random in the whole population, then the individual with the rarest phenotype is kept. The tournament is performed every time there is an individual to select. Thus it is performed twice to select the 2 parents of a new individual to be created.

## Crossover and mutation

We used the adaptive Cauchy mutation [28, 29] and SBXCrossover [30].

## Elitism

The elitism stage limits the growth of the population size. After the evaluation of the new offsprings, each of them is attributed to an archive cell corresponding to its phenotype. In cells where there is more than 1 individual (considering new offsprings as well as individuals of the

parent population), one is selected randomly and the others discarded. The individuals that remain form the new population. This is comparable to a *plus* selection scheme, denoted  $(\mu + \lambda)$  where  $\mu$  is the population size and  $\lambda$  the number of offsprings, with the particularity that the population size grows since a new individual located in a cell with a 0 hit count can be incorporated into the population without replacing any other.

## Stop condition

PSE runs continually. It is up to the user to decide when to stop the exploration. The decision can be based on the evolution of the number of distinct patterns (or volume discovered, i.e. the number of cells in the archive whose hit-count is greater than 0), for example, when the rate of pattern discoveries slows down. Caution must be taken about interpreting this as a sign that all or most patterns were discovered. There are two possible interpretations to the slowing down of the rate of discoveries: either the model cannot exhibit any other pattern (according to the discretisation scheme) than the ones already discovered, or the remaining patterns are located in a region of the search space that is very hard for the exploratory process to access. The latter possibility is the consequence of non-linearity between the parameter space and the pattern space. This uncertainty is inherent to evolutionary algorithms, but it is also what constitutes their strength: the ability to continually look for better solutions (in the case of objective-based evolutionary algorithms) or to search for diversity (in the open-ended objective-less case) in absence of any a priori information about the relationship between the search space and the output or pattern space. They are best suited for the cases where little such information is available to the modeller, which is common in computer-based models of complex systems.

## Parallelism

Evolutionary algorithms are naturally adapted to parallel computing since individuals evaluations are independent from one another. In the experiments below, we used two different parallelisation strategies: a parallelised steady state model and an island steady state model. (The first strategy made easier to control and record the number of model evaluations performed in time, but was slower on the EGI grid (see section [Software and hardware infrastructure](#)). The second strategy was more efficient at exploiting the grid's computing resources and was thus used for the most computationally intensive experiment.)

With the standard steady state model, only one individual is created at each generation. For parallelisation, we had a pool of  $\lambda$  individuals continually evaluated in parallel. Every time an evaluation is done, the corresponding individual and the current population go through the stages of elitism, selection, crossover and mutation to generate a new individual fed back to the pool for evaluation. This is close to the standard steady state model,  $(\mu + 1)$ , with the difference that when a new individual is created, some of the individuals that were created before it are still in the evaluation pool and weren't yet sent back to the population.

The island steady state model works by splitting the global evolution into several smaller evolutions (islands) which can be run in parallel. A global population and global hit-map are created on the main computing node. When an island is started on a distant node, it is initialised with a random subset of the global population. It then runs the complete PSE algorithm with the standard steady state model  $(\mu + 1)$  for a fixed amount of time. When it completes, its final population is used to update the global hit-map and is merged back into the global population with the same elitism operator as described above. Subsequent islands are then initialised with populations sampled from the updated global population. A fixed number of islands are run in parallel. When one finishes, a new one is started.



## Usage

To use PSE on a given model, the following information must be specified:

- which model parameters constitute the genotypic space (some model parameters may be excluded from the search space by fixing their values or initialising them randomly),
- a set of measures describing the simulation output, constituting the phenotypic space
- bounds on the parameters,
- discretisation step size on the phenotypic dimensions,
- $\mu$ , the size of the initial population,
- the probability of reevaluating a selected individual without performing any crossover and mutation,
- if the parallelised steady state model is used,  $\lambda$ , the size of the evaluation pool,
- if the island steady state model is used, the number of islands run in parallel and the population size for each island.

## Software and hardware infrastructure

PSE was developed in Scala within the MGO library, an open source library for evolutionary algorithms. The source code is available as a git repository (<https://github.com/openmole/mgo>).

The models presented below were also developed in Scala within the Simpuzzle project, and also available as a git repository (<https://github.com/ISCPIF/simpuzzle>).

The simulation experiments described below were deployed on the European Grid Infrastructure through the *biomed* virtual organisation.

Parallel deployment of the simulations on the grid was performed with OpenMOLE (<http://www.openmole.org/>) [31], an open source software developed at ISC-PIF dedicated to the execution of simulation experiments on distributed computing environment (multicore machines, clusters, grids). MGO is tightly integrated to OpenMOLE, which allowed to easily use PSE within OpenMOLE to drive the experiments.

The script files, models and instructions necessary to reproduce the experiments presented below are available in the github repository <https://github.com/ISCPIF/PSEExperiments>.

Developers interested in using PSE without OpenMOLE should refer to the MGO library mentioned above. Refer to the last section of the README file in the PSEExperiments github repository for instructions to get started.

## Test case: collective motion

The study of collective motion, or flocking, is the study of how self-propelled and locally interacting particles can organise to move together as a whole. What is particularly intriguing about these systems is their capacity to produce collective motion and complex collective behaviours like avoiding obstacles without any leader. These phenomena have been extensively studied by physicists starting with Vicsek [32, 33], and others since then, both analytically and in simulation [34].

The literature contains precise descriptions of the collective behaviour of flocking models [6]. Knowing what behaviours were to be expected made it a good test case for our method.

We tested our Pattern Search Exploration method on the model described below. In order to compare the method to some famous a priori experimental designs, we also used a LHS, a regular grid sampling, and the Sobol sampling method.

### The flocking model and simulation

We implemented a version of Reynolds' famous flocking model [35] inspired by the flocking model available in Netlogo's model library [36, 37].

The entities called boids are located in a 2D toroidal environment. Each is described by a position and a heading. For an environment of width  $E_w$  and height  $E_H$ , the boid  $i$  has a position  $(x_i, y_i) \in [0, E_w) \times [0, E_H)$  and a heading  $h_i \in [-0, 2\pi)$ .

Each boid tries to avoid collision with nearby flockmates while matching their heading and staying close to them. The collision avoidance mechanism has precedence over the other two.

All boids positions and headings are updated simultaneously from one time step to another, avoiding any effect of the order in which new states are computed. At each time step, a new heading is computed for each boid based on its current heading and the state of its neighbours, i.e. other boids within radius  $rVis$ . Boids then step forward by a fixed distance  $stepSize$ .

If the distance of boid  $i$  to its nearest neighbour  $j$  is less than  $minSep$ , then  $i$  only tries to avoid collision. It turns away from  $j$ 's heading, with a maximum angle turn of  $maxTurnS$ . Its new heading is given by:

$$h_i(t + 1) = h_i(t) + T(A(h_i(t), h_j(t)), maxTurnS).$$

Where  $A(h_i, h_j)$  gives the angle from  $h_i$  to  $h_j$  such that  $(h_i + A(h_i, h_j)) \bmod 2\pi = h_j$ . The function  $T$  keeps the turn angle absolute value less than or equal to  $maxTurnS$ :

$$T(a, m) = sign(a) \times min(abs(a), m), \quad m \geq 0.$$

If boid  $i$  has no neighbour that is too close, it will simultaneously try to match its flockmates average heading and turn towards their average position. In each component (match flockmates headings and turn towards flockmates) the turn is limited by a maximum angle respectively of  $maxTurnA$  and  $maxTurnC$ :

$$h_i(t + 1) = h_i(t) + T(\langle h_j(t) \rangle_{j \sim i}, maxTurnA) + T(\theta(\langle x_i - x_j \rangle_{j \sim i}, \langle y_i - y_j \rangle_{j \sim i}), maxTurnC).$$

$j \sim i$  is the set of particles  $j$  within distance  $rVis$  of  $i$ .  $\theta(x, y)$  gives the  $\theta$  component of the point  $(r, \theta)$  in polar coordinates that corresponds to the point  $(x, y)$  in Cartesian coordinates.

The population is constituted of  $N_b$  boids which are initialised randomly at the beginning of a simulation.

The model has 9 parameters (see Table 1). In the following simulations, the four first are fixed to the values displayed in the table. The model has thus five free parameters which will constitute the search space.

Simulations are run for 1000 time steps, both to give reasonable time for flocks to form and avoid too long simulation time.

### The pattern space

Grégoire, Chaté and Tu [6] proposed five qualitatively different behaviours a flock can exhibit: gas, moving solid, stationary solid, moving liquid and stationary liquid. In the gas phase the particles do not coalesce into flocks but move independently. In the solid phase, particles keep a fixed position relative to one another in time. In the liquid phase, particles move relative to

**Table 1. Parameters for the flocking model.**

Parameter	Value	Description
$E_w$	32	Environment's width
$E_H$	32	Environment's height
$N_b$	128	Number of boids
<i>stepSize</i>	0.05	Boids speed
<i>rVis</i>		Neighbourhood radius (vision)
<i>minSep</i>		Minimum separation distance allowed to avoid collision
<i>maxTurnS</i>		Maximum turn allowed to avoid collision
<i>maxTurnA</i>		Maximum turn allowed to align with flockmates
<i>maxTurnC</i>		Maximum turn allowed to get closer to flockmates

doi:10.1371/journal.pone.0138212.t001

one another within the flock, as seen in schools of fishes and flocks of birds. We used these five kinds of behaviours as landmarks to test our method.

They use three order parameters to describe a flock's behaviour, described below, which we use to constitute the pattern space. They describe the formation of flocks, liquid/solid phases, and the flocks' velocity.

The gas phase corresponds to particles not forming a cohesive flock. "A cohesive flock is one for which  $n$ , the size of the largest cluster, is of order  $N$ , the total number of particles" [6]. As the first dimension of the observable space we can take the size of the biggest cluster. The clusters were computed after 1000 iterations of the model. They are formed by connecting together the particles which are within a distance  $rVis$  of one another. The clusters are the connected components of the resulting graph.

The solid and liquid phases differ by how much the particles move relatively to one another during the simulation, that is how the distance between initially neighbouring entities at time  $t$  varies at  $t + T_d$  for large  $T_d$ . To capture this phenomena, Grégoire, Chaté and Tu [6] use the following relative diffusion measure:

$$\Delta \equiv \left\langle \frac{1}{n_i} \sum_{j \sim i} \left( 1 - \frac{r_{ij}^2(t)}{r_{ij}^2(t + T_d)} \right) \right\rangle_{i,t}$$

where  $n_i$  is the number of neighbours of particle  $i$  at time  $t$ , and  $r_{ij}$  the distance between particles  $i$  and  $j$ . In  $j \sim i$ , the  $j_s$  are all the neighbours of  $i$  in the Voronoi sense.  $\Delta \sim 1$  in the liquid phase and  $\Delta \sim 0$  in the solid phase.

Rather than averaging over all  $t$ , which is computationally costly, we limited  $t$  to 5 different values towards the end of the simulation, i.e.  $t \in \{1000 - T_d, 1000 - 2T_d, 1000 - 3T_d, 1000 - 4T_d, 1000 - 5T_d\}$ . The environment being toroidal, too large values of  $T_d$  would have compromised the distance measures. When two initially neighbouring particles move in different directions, their distance starts shrinking after growing beyond half the environment's linear size. We set  $T_d$  so that an individual could not travel more than a fourth of the environment linear size,  $T_d = \min(E_w, E_H)/(4 \times stepSize)$ . It remains possible, due to the fact that neighbours are usually separated by a non null distance, that initially neighbouring individuals moving away from each other will cross that distance, leading to smaller  $\Delta$  than should be observed. This pattern however should only concern cases where boids do not remain in cohesive flocks, i.e. the gas phase. The problem should thus not interfere when cohesive flocks do form, which is when  $\Delta$  will be of interest to distinguish the liquid phase from the solid phase.

**Table 2. Bounds on the flocking free parameters domains.**

	<i>rVis</i>	<i>minSep</i>	<i>maxTurnS</i>	<i>maxTurnA</i>	<i>maxTurnC</i>
min	0	0	0	0	0
max	32	32	$\pi$	$\pi$	$\pi$

doi:10.1371/journal.pone.0138212.t002

Finally, in order to distinguish between the stationary and moving phases, we measured the average velocity  $V$  of the particles towards the end of the simulation. The velocity at  $t$  was measured by the average distance between each individual's position at  $t$  and  $t + T_v$ , where  $T_v = \min(E_W, E_H)/(2 \times stepSize)$ .

$$V = \left\langle \frac{1}{N} \sum_i d_i(t, t + T_v) \right\rangle_t,$$

where  $d_i(t, t + T_v)$  is the euclidean distance travelled by particle  $i$  between  $t$  and  $t + T_v$ . Here again,  $t \in \{1000 - T_v, 1000 - 2T_v, 1000 - 3T_v, 1000 - 4T_v, 1000 - 5T_v\}$ .

### Pattern Space Exploration configuration

The variables constituting the genotypic space are the free parameters of the model, and their domain bounds are given in Table 2. The boids positions and headings are initialised randomly. Individual evaluation was done with 100 replications. The phenotype is the median of each 3 variables of the pattern space over the replications. Table 3 gives the discretisation step size of their respective domains. Additionally, the domain of exploration for  $\Delta$  was bounded with a minimum of -1. Simulations whose  $\Delta$  value were beyond it were assumed to have  $\Delta = -1$ . Since  $\Delta$  itself has no lower bound, this artificial bound was set to avoid pushing the exploration below -1.

The initial population size parameter  $\mu$  was set to 5000, and the probability of reevaluating an individual (rerun all replications on a selected point without crossover or mutation) to 0.01. We let the exploration run until the total number of evaluations reached 100000 (i.e. 10 million simulations, since there are 100 replications per evaluation). The parallelised steady state model was used. The size of the evaluation pool  $\lambda$  was set to 5000.

In order to test the robustness of the method regarding stochasticity, the exploration was replicated 10 times.

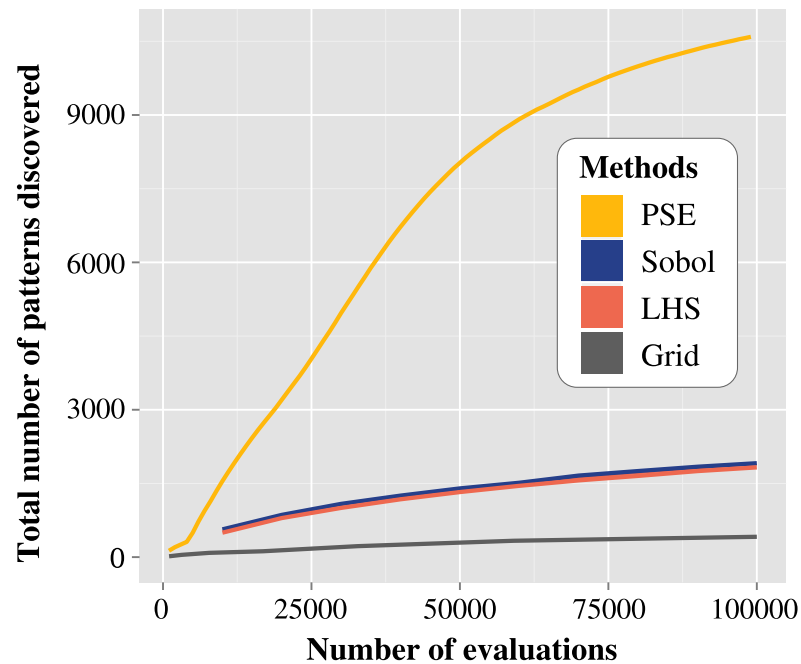
### Benchmark of PSE against a priori samplings

We compared PSE's ability to reveal the diverse patterns a model can produce, using the flocking model, to common a priori sampling methods: a regular grid, LHS and Sobol's sampling.

**Table 3. Discretisation step size of the observables for the flocking model.** Min and max values give the theoretical bounds for each observable. Between parentheses is given an artificial bound that was set for  $\Delta$  in the experiments. Any lower value encountered in a pattern were set back to -1 in the phenotype. This trick allowed us to focus the exploration on regions of interest.

	<b>Biggest cluster size</b>	<b>Relative diffusion (<math>\Delta</math>)</b>	<b>Average velocity (<math>V</math>)</b>
step	1	0.2	0.1
min	1	$-\infty$ (-1)	0
max	128	1	0.5

doi:10.1371/journal.pone.0138212.t003



**Fig 3. Comparison of the rate of pattern discovery in time for the flocking model.** Number of patterns discovered as a function of the number of evaluations with the different methods tested (the model initialisation is stochastic, therefore 1 evaluation = 100 simulations). For PSE and LHS, averages over 10 replications are represented. Standard deviations were too small to be represented.

doi:10.1371/journal.pone.0138212.g003

Those produce points in the parameter space. As with PSE, for each point, 100 replicated simulations were run, and the median for each pattern variable was computed.

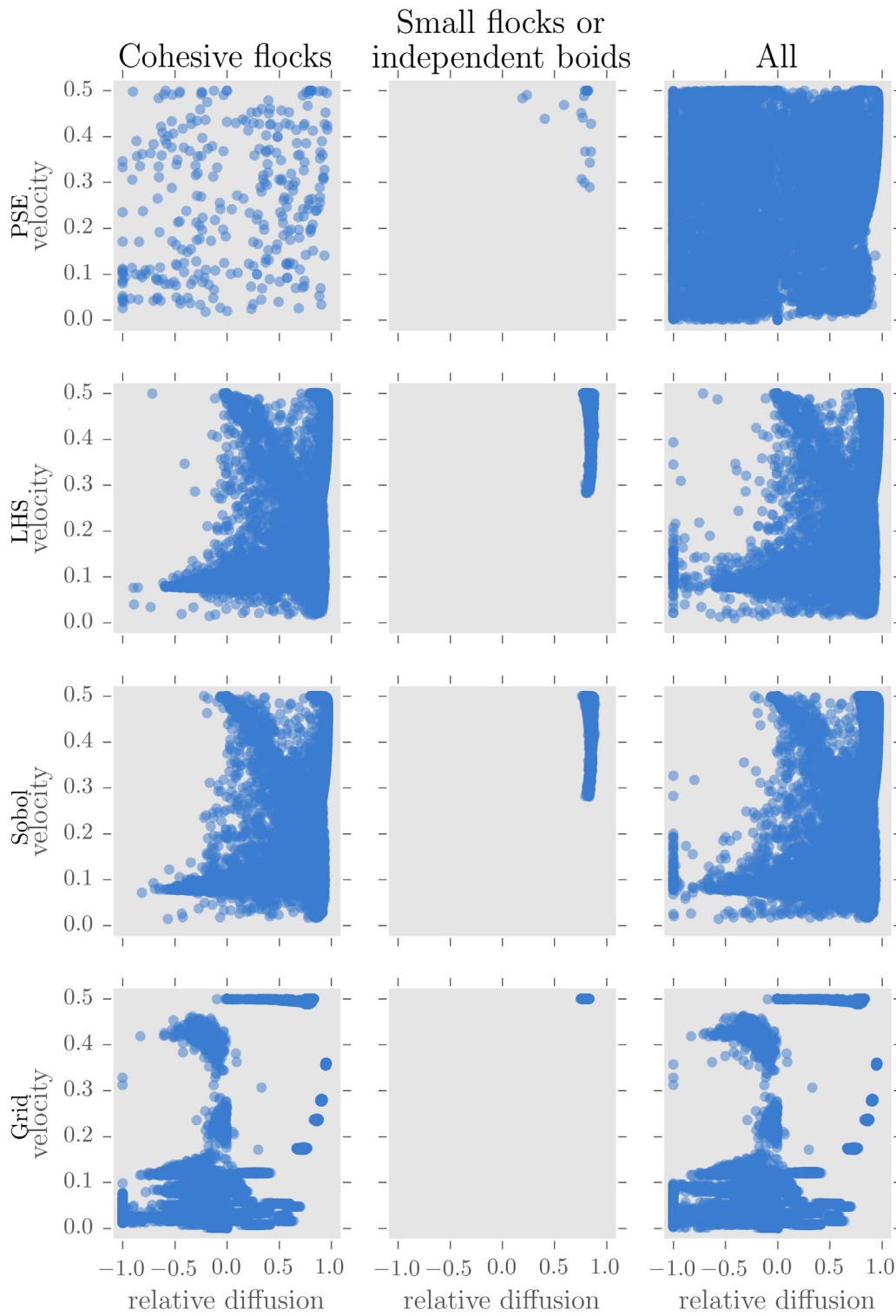
**Regular grid.** In the regular grid sampling, each parameter domain was sampled at regular intervals. A sampling was then produced by all the combinations of values for each parameter. We produced 6 samplings respectively containing  $i^5$  points,  $i$  varying from 4 to 10 being the number of samples per parameter domain.

**LHS and Sobol.** These methods are multidimensional low-discrepancy sampling methods. LHS is stochastic and Sobol is deterministic. For each, we generated 10 samplings respectively containing  $i \times 10^4$  points, with  $i$  varying from 1 to 10. The LHS sampling was replicated 10 times.

## Results

**Patterns discovered with PSE.** The progression of the number of patterns discovered slows down after 60000 evaluations (Fig 3). Out of the 100000 evaluations, 95% of the patterns were discovered within the 81400 first evaluations on average (sd. 663).

The five different kinds of patterns enumerated previously were found. In Fig 4, the top row shows the patterns found by PSE. The left column shows patterns where large cohesive flocks form, and the middle column those where particles move independently or in small groups. The latter correspond to the gas phase. In the former, we can distinguish the 4 other kinds of patterns.



**Fig 4. Patterns discovered by PSE and 3 a priori samplings using the flocking model.** Plots on the left show the results of simulations where big clusters of at least 125 boids formed, plots in the middle those where the clusters were constituted by 2 individuals at most, and plots on the right show the results of all simulations.

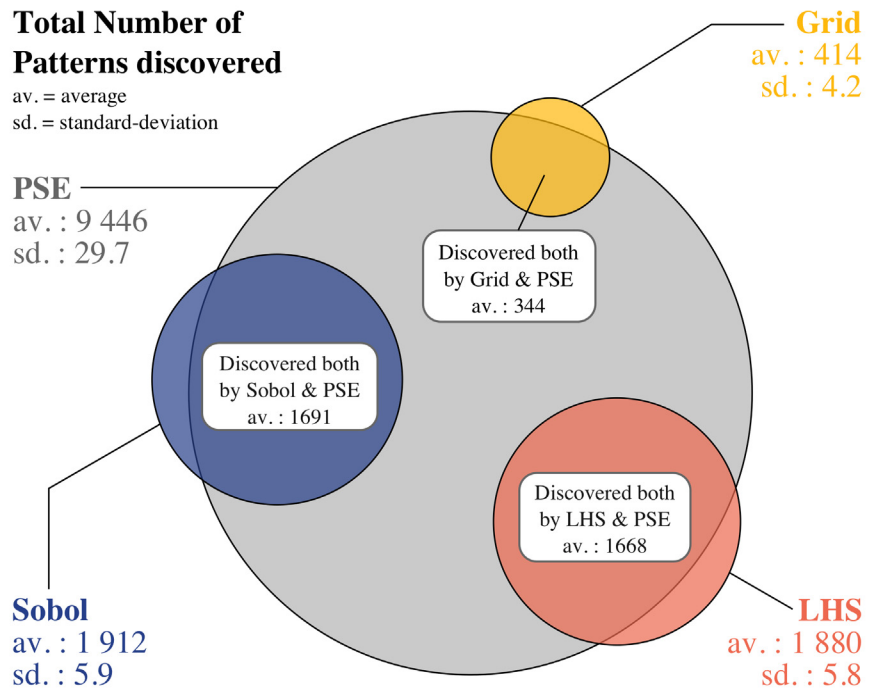
doi:10.1371/journal.pone.0138212.g004

In the top left hand plot, patterns with a relative diffusion ( $\Delta$ ) close to 0 corresponds to the solid state, and those close to 1 correspond to the liquid state. The exploration also found negative values which denote flocks contraction. On the velocity axis, we can distinguish between moving (0.5 is the velocity of a particle moving straight) and stationary states. We can see that the exploration found a range of patterns from stationary to moving and from liquid to solid to contracting motion.

These results were similar across the 10 replications of PSE (see S1 Fig).

**Comparison of PSE to other a priori samplings.** As seen in Fig 3, PSE was the fastest and most efficient method in terms of number of distinct patterns discovered. PSE found many patterns which were not discovered by the other methods, and only few were discovered by the other methods that were not discovered by PSE, as seen in Fig 5.

Fig 4 offers a visual comparison of the patterns discovered by the different methods. PSE found a bigger variety of behaviours for cohesive flocks as well as small flocks. The LHS and Sobol samplings had trouble finding cohesive flocks with negative relative diffusion and high velocity (see S2 Fig for the results of each LHS replication). The regular grid sampling also missed many patterns of positive relative diffusion which were largely discovered by all other methods.



**Fig 5. Number of patterns discovered by PSE and each compared method using the flocking model.** Quantities are averaged over the experiments replications for stochastic methods. Intersections between the a priori samplings are not shown.

doi:10.1371/journal.pone.0138212.g005

## Use case: exploration of the patterns of hierarchisation and population growth of a system of cities with the model MARIUS

In an empirical case study about the evolution of the system of cities in the post-Soviet space, the method proved useful to explore the behaviour of a large agent-based model (in terms of the number of agents (1145) and running time), and to give an answer to some thematic and theoretical questions regarding the evolution of this system as modelled within the model MARIUS.

### Thematic incentives for Pattern Space Exploration

The objects of investigation of the social sciences in general, and the geographical study of systems of cities in particular, face the problem of the impossibility to run experiments *in vivo* and to reproduce an observed evolution under different conditions. However, thinking of societies as complex systems imply to consider parts of their trajectories and configurations as contingent, and to acknowledge that the observed outcome is just one out of many different possible scenarios [38, 39]. We could even say that “to know a society and a geography is to know how it could be different than it is.” [40]

This is why computer-based modelling and simulation *in silico* provide interesting tools to explore the possible outcomes of a given society and geography under changing sets of conditions [41], and in particular, this is why the PSE tool is a good way to explore the diversity of possible trajectories that the modelled systems can reach according to changes in the parameters.

In the case of MARIUS, which aims at simulating interactions and growth patterns of cities of the Former Soviet Union, we wanted to explore the possibility of unprecedented situations. Almost all the systems of cities that we are able to monitor via census data in the world since the invention of censuses are growing systems, and most of them have exhibited a tendency towards hierarchisation—the accentuation of the degree of inequality regarding cities populations (a city population and its rank in the system are related by a power law, meaning hierarchised systems have a few large cities and many small ones, as in the canonical Zipf's Law) [42]. However, post-Soviet cities, and Russian cities even more, have been experiencing an unprecedented process of demographic shrinkage since the fall of the USSR, while persisting in the hierarchisation of the system [43, 44].

MARIUS was developed to reproduce the particular structure of the Soviet system of cities with a parsimonious set of generic rules. A study showed its ability to reproduce the cities historical hierarchical structure evolution from 1959 to 1989 [27], i.e. hierarchisation of the cities in a context of population growth. The corresponding model parameters calibrated values are shown in Table 4. In this paper, we use PSE to find out which possible degrees of hierarchisation and growth the model can produce. Four main types of outcomes can be expected, the first three of them were already confirmed by historical data as mentioned above:

- hierarchisation in a growth context (e.g. United-States in the XXth century, Russia between 1959 and 1989),
- hierarchisation in a shrinkage context (e.g. Russia in the 1990s and 2000s, Germany in the 2010s),
- equalisation in a growth context (e.g. China in the 1960–70s),
- equalisation in a shrinkage context.



**Table 4. MARIUS parameters.** Bounds for the experiments and calibrated values from [27] reproducing Russia's cities hierarchisation and growth during the late XXth.

parameter	min	max	calibrated value
<i>economicMultiplier</i>	0	100	0.3438093442
<i>sizeEffectOnSupply</i>	1	2	1.0017563880
<i>sizeEffectOnDemand</i>	1	2	1.0792607803
<i>wealthToPopulationExponent</i>	0	2	0.3804356044
<i>distanceDecay</i>	0	10	0.6722631615
<i>populationToWealthExponent</i>	1	10	1.0866012754
<i>bonusMultiplier</i>	0	1000	197.9488907791
<i>fixedCost</i>	0	1000	0.2565248068

doi:10.1371/journal.pone.0138212.t004

## MARIUS

The full ODD description of the model is available in the [S1 Appendix](#) and on github (<https://github.com/ISCPiF/marius-method/blob/master/ODD-MARIUS.pdf>). The following is a brief overview.

The MARIUS model includes basic economic rules of interurban interactions. The system is made of collective agents: the 1145 agglomerations of more than 10 000 inhabitants in 1959, localised via actual latitudes and longitudes in former Soviet Union. It is initialised with empirical populations and a fictive wealth quantity estimated as a superlinear function of population (i.e. with the parameter *populationToWealthExponent* being greater or equal to 1).

At each step:

1. Each city updates its economic variables according to its population: supply and demand, modelled as superlinear scaling quantities. Two parameters, *sizeEffectOnSupply* and *sizeEffectOnDemand* shape the non-linear advantages in productivity and consumption per inhabitant according to the city size. The higher they are, the more big cities are economically productive and consumptive per capita. The parameter *economicMultiplier* adjusts these quantities to a fictive unit of wealth.
2. Each city interacts with the others according to the intensity of their interaction potential (a function of the masses of the two cities and the distance between them, depending on a parameter *distanceDecay* representing the role of distance on interaction potential). For two distinct cities A and B, an interaction consists in confronting A's supply to B's demand, resulting in a potential transaction of goods sent from A to B.
3. Potential interactions are selected if the potential exchange exceeds a profitability threshold *fixedCost* representing the fixed costs of interurban exchanges non proportional to distance (the 4T costs of [45]: transaction, transportation, temporal, tariff and non-tariff costs). This mechanism reflects the fact that pairs of small and distant cities are not supposed to interact for very small amounts. That way, cities do not necessarily exchange with all other cities in the system.
4. Each city updates its wealth according to the results of the transactions in which it was committed and gets an economic impulse from the amount and diversity of exchanges. This reward of interactions relies on a parameter called *bonusMultiplier*, which represents the magnitude of exchanged externalities (information, knowledge, acquaintance, etc.).

- Each city updates its population according to its new resulting wealth. The conversion is differentiated according to the size of the city's population, with a scaling exponent *wealthToPopulationExponent*. The economic literature is not definitive about the range of values for this parameter [46, 47]. Below a value of 1, the same gain in wealth generates proportionally more population growth in smaller cities. Over 1, the same gain in wealth is demographically greater for larger cities.

The date used for initialisation is 1959, as it refers to a census date that opens a period of stability in the Soviet Union and a generic evolution of urbanisation. Each simulation runs for 30 time steps corresponding to one year each, therefore leading to a simulated 1989.

### The pattern space

The pattern space for MARIUS is 3-dimensional. It was designed to reflect first of all the hierarchisation and population growth of the system. Hierarchisation was measured as the slope of the size-rank distribution at the end of the simulation, i.e. the exponent of the power law relating a city size to its rank in the system. Population growth was measured as the difference between the final total population and the initial one. To distinguish between simulations where population growth is monotonic from those where it is more complex, we added as a third variable the number of times the sign of the population growth changed during the simulation.

### Pattern Space Exploration configuration

The model has eight free parameters. Their domain bounds are given in Table 4.

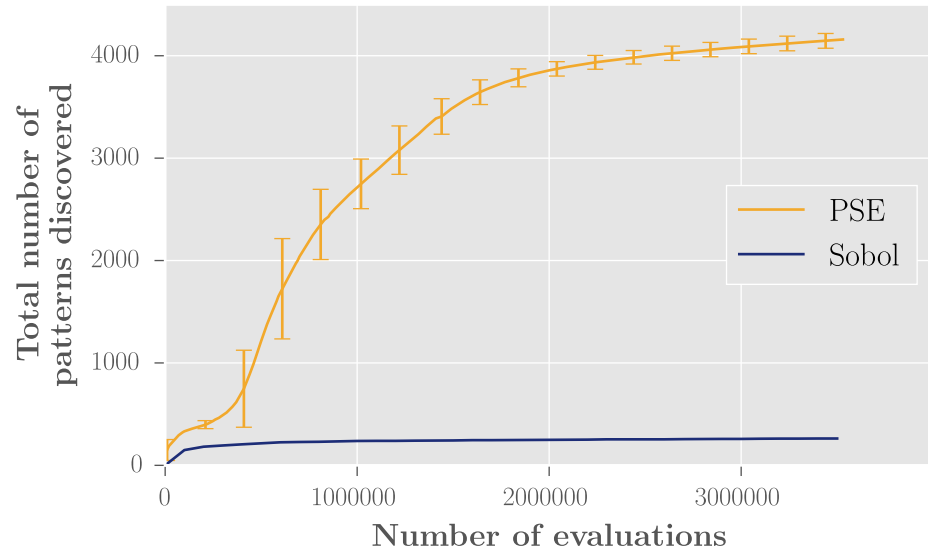
The model is deterministic. The phenotypic space corresponds to the pattern space described in the previous section. Each dimension is discretised according to Table 5. The phenotypic variables were bounded to focus the exploration on intervals of interest. The population growth was upper bounded to 500 million, the inversion count was upper bounded to 3, and the rank-size slope (hierarchisation) was lower bounded to -10. Simulations whose pattern values went beyond these were assumed to have a phenotypic value of the corresponding bound.

First, we ran a “long run” experiment where we let PSE run for a total of approximately 36 million model evaluations. We executed a total of 200000 islands for two hours each, distributed on 5000 computing units. One model evaluation takes about 40 seconds. The whole run took 400000h in cumulative computation time. The actual computation time was reduced to about a week thanks to the computational capabilities of the EGI grid. In this first experiment, the exploration was not replicated. To compare the rate of discovery between PSE and Sobol, we performed a second experiment involving shorter explorations using the parallel steady

**Table 5. Pattern variables for MARIUS.** The step column gives the discretisation step size. Min and max values give the theoretical bounds for each observable. Artificial bounds were set on some parameters to focus the exploration on regions of interest. They are given inside parentheses. Calibrated values are from [27], reproducing Russia's cities hierarchisation and growth during the late XXth.

	step	min	max	calibrated value
Rank-size slope	0.1	$-\infty$ (-10)	0	-1.151
Population difference	5000	$-0.83544 \times 10^8$	$+\infty$ ( $500 \times 10^8$ )	65554000
Inversion count	1	0	28 (3)	0

doi:10.1371/journal.pone.0138212.t005



**Fig 6. Number of patterns discovered by PSE and Sobol using the model MARIUS.** Abscissa gives the number of model evaluations. The PSE curve displays the average and standard deviation over 10 replications. PSE was much more efficient than the Sobol sampling.

doi:10.1371/journal.pone.0138212.g006

state method. PSE was replicated 10 times and each exploration (the 10 PSE replications and the Sobol sampling) was run for 3.5 million model evaluations.

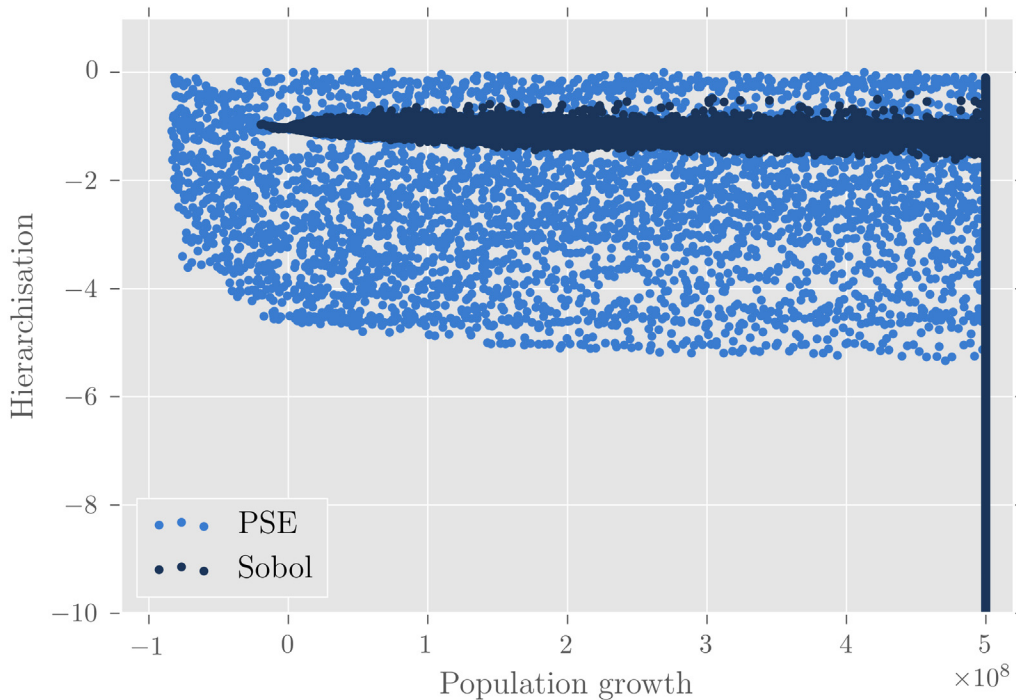
## Results

During the first long run experiment, a total of 4555 distinct patterns were discovered, and 95% of them were discovered within the first 48000 hours (cumulative time of the simulations running in parallel). The exploration stopped making new discoveries after about 150000 hours (13.5 million model evaluations). Fig 6 shows the evolution of the number of patterns discovered during the second experiment. The comparison between PSE and the Sobol sampling shows that PSE was much more efficient at discovering diverse patterns for the model MARIUS's (Figs 6 and 7).

The patterns found during the long run exploration are presented in more details in Fig 8 and analysed below. The results are shown for 0 (monotonic population growth), 1 and 2 inversions.

The grey spans correspond to a realistic behaviour of empirical systems of cities with respect to the total growth of urban population (5% per year being the maximum observed in recent China, [48, 49]) and to the hierarchy of their size distribution (the maximum empirical range for the exponent of the rank-size distribution in the recent world being comprised between 0.8 in Egypt and 1.5 for 70 countries in the last 30 years [50]).

The dots on the far right which go below a hierarchisation value of -6 correspond to simulations where population growth is greater than 500 million. As explained above, simulation where population growth is greater than 500 million were assumed to have a growth of 500 million, and are shown at this value on the plot (likewise for the hierarchisation which was bounded at -10). Preliminary experiments not shown here revealed that population growth can



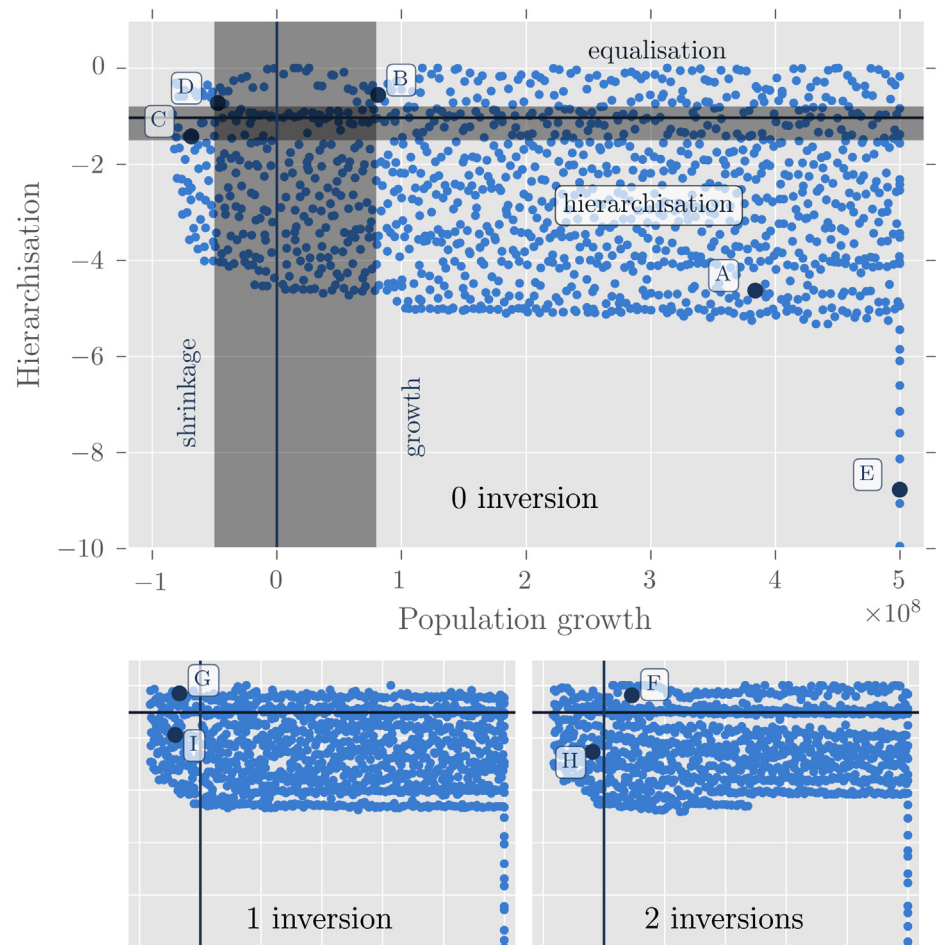
**Fig 7. Patterns discovered by PSE and the Sobol sampling for the model MARIUS.** Only one replication of PSE is shown. Both methods were run for 3.5 million model evaluations. Each blue dot corresponds to a pattern. PSE found a more diverse range of patterns than the Sobol sampling.

doi:10.1371/journal.pone.0138212.g007

get far above 500 million, and hierarchisation can get far below -10, and the number of inversions up to 28 (the maximum that can happen in the 30 steps of the simulation).

Indeed, the 4 kinds of expected behaviours (hierarchisation and growth, hierarchisation and shrinkage, equalisation and growth, equalisation and shrinkage) were found within the empirically plausible bounds. But many patterns were also found to be distributed outside of the grey spans. These patterns are among the potential falsifiers or potential predictions of the model, and are what PSE was designed to discover. They constitute the basis on which the model’s assumptions can be revised. We picked nine and inspect them in more details below. They are labelled from A to I in Fig 8, the evolution of their rank-size distribution is given in Fig 9, and their parameter and pattern values are given in Table 6.

Patterns A and B contrast mainly in the behaviour exhibited by small cities (Fig 9). In A, hierarchisation is performed by their continuous loss of population in favour of the 100 largest cities, while in B, on the contrary, growth of total urban population is due exclusively to the rise of small cities, resulting in an unprecedented equalisation pattern. The distribution of growth with relation to size is controlled by the parameter *wealthToPopulationExponent*, 0.62 in A and 0.23 in B, meaning that the same gain of wealth in two cities attracts more people to the smallest cities in both configurations, but to a larger extent in B, resulting in this equalisation of city sizes. Pattern A represents an unlikely configuration because hierarchisation is usually achieved by a relatively higher growth rate for large cities, and not through a dramatic shrinkage of the small ones as in A. B is less unlikely but it goes against recent observation of a larger attraction of cities with large populations. It could be a prediction of evolution in a context of high costs of congestion and preference for small urban clusters.

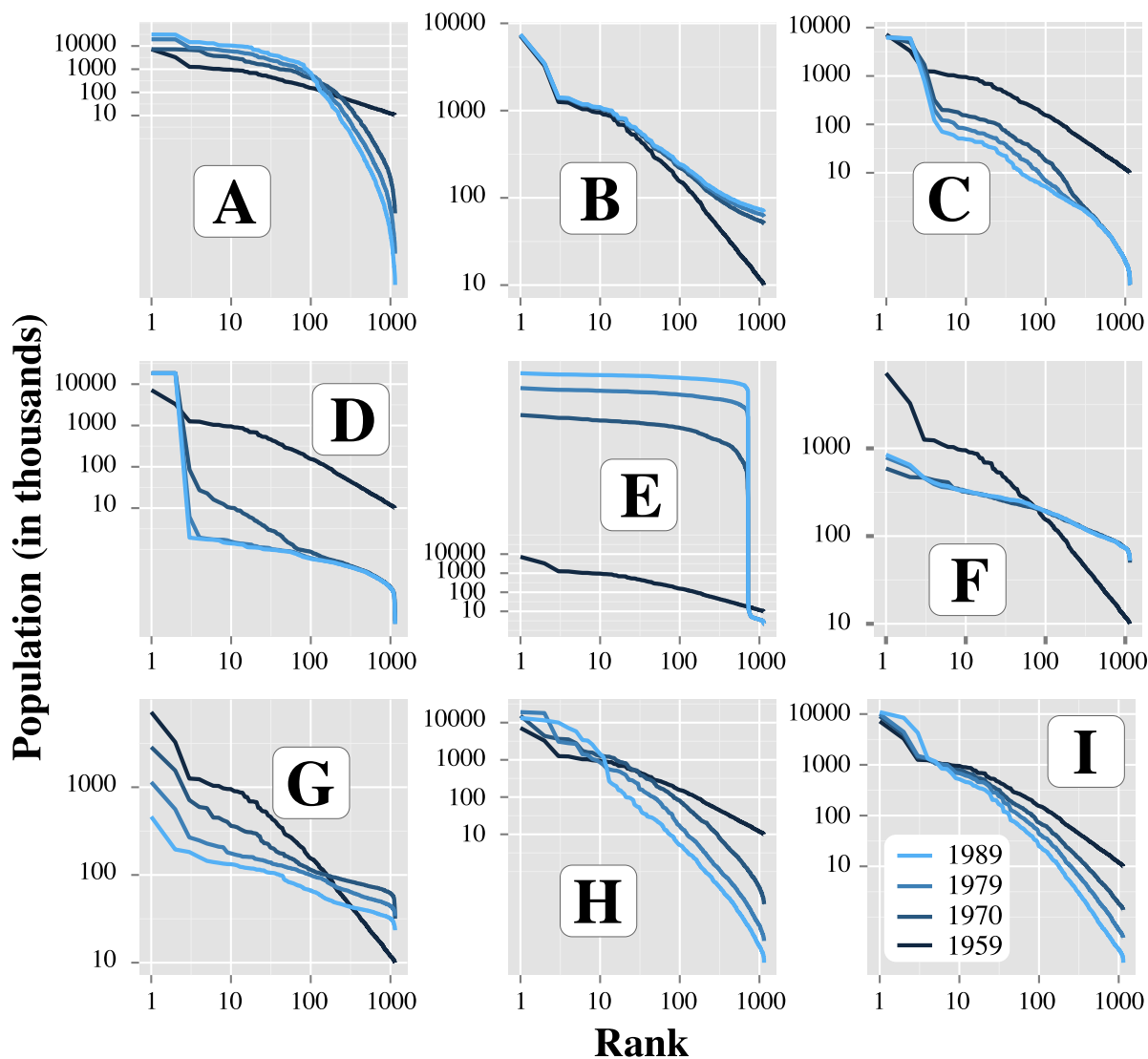


**Fig 8. Patterns discovered for the model MARIUS with PSE.** The vertical and horizontal lines show the initial state of the system. The grey intervals represent plausible values for each dimension. Letters A-I give the behaviours that are inspected in more details in Table 6, Fig 9 and in the Results section.

doi:10.1371/journal.pone.0138212.g008

In a context of shrinkage, pattern C and D both appear very unlikely, at least, in such a short period of time. First of all, those extreme simulations correspond to a loss of half the system’s population in 30 years. C corresponds to a shrinkage process coupled with hierarchisation, while in D, cities are less differentiated on average as the simulation goes. In both patterns, the majority of cities experience a loss in their population equivalent to one or two orders of magnitude within the first eleven years (1959–1970), and then stagnate around this value. This process is very unrealistic. These patterns must be considered falsifiers of the model. The model assumptions responsible for them need to be re-examined. On the other hand, patterns C and D appear similar in the way the two largest cities (Moscow and Saint Petersburg in the initialisation) end the simulation in a macrocephalic position. This suggestion of resilience for the largest cities of a system is consistent with recent observed trends in shrinking city systems and prospective scenarios of shrinkage management [43].

Finally, in a scenario of continuous growth (with no inversion), Pattern E appears as a most unrealistic extreme case. It results from the very high value of the *bonusMultiplier* parameter (412), that over-values any exchange taking place with another city with respects to the



**Fig 9. Evolution of the rank-size distributions in the nine inspected simulations of the model MARIUS.** The rank refers to the position of the cities sorted by decreasing population size.

doi:10.1371/journal.pone.0138212.g009

production value. This is an example of the pattern exploration highlighting unreasonable parameter bounds.

Pattern F to I involve one or two inversions of growth. We explored these patterns for prospective purposes, since reversion of urban growth has happened twice in Russia during the last twenty years. By exploring those patterns, we aim at evaluating the interactions between reversion of growth and hierarchisation. Those four patterns show that growth inversions have minor influence on the overall patterns (in F and G: equalisation of city sizes and in H and I: hierarchisation).

F and H have in common two growth inversions and the deactivation of the exchange criteria (*bonusMultiplier* and *fixedCosts* = 0), which means that an exchange within the city has the

**Table 6. Parameter (italics) and pattern (bold) values for the 9 inspected patterns.**

	A	B	C
<i>economicMultiplier</i>	0.002108	0.248015	0.004238
<i>sizeEffectOnSupply</i>	1.535866	1.392815	1.911325
<i>sizeEffectOnDemand</i>	1.088266	1.180051	1.223311
<i>wealthToPopulationExponent</i>	0.615699	0.228982	0.494603
<i>distanceDecay</i>	2.516997	0.395674	0.859162
<i>populationToWealthExponent</i>	1.770787	1.016049	1.605315
<i>bonusMultiplier</i>	0.122952	169.159662	0.080951
<i>fixedCost</i>	0	0	0.289442
<b>Rank-size slope</b>	-4.62	-0.56	-1.42
<b>Population difference</b>	$383 \times 10^6$	$81 \times 10^6$	$-68 \times 10^6$
<b>Inversion count</b>	0	0	0
	D	E	F
<i>economicMultiplier</i>	0.044799	0.022930	0.199871
<i>sizeEffectOnSupply</i>	1.505378	1.659717	1.099736
<i>sizeEffectOnDemand</i>	1.173295	1.617497	1.544439
<i>wealthToPopulationExponent</i>	0.796531	0.505149	0.759320
<i>distanceDecay</i>	0.846435	0.428968	0.798650
<i>populationToWealthExponent</i>	1.271591	1.118053	1.348373
<i>bonusMultiplier</i>	0	412.164462	0
<i>fixedCost</i>	0.021403	0.791581	0
<b>Rank-size slope</b>	-0.73	-8.77	-0.38
<b>Population difference</b>	$-47 \times 10^6$	$500 \times 10^6$	$46 \times 10^6$
<b>Inversion count</b>	0	0	2
	G	H	I
<i>economicMultiplier</i>	0.009592	0.247120	0.000592
<i>sizeEffectOnSupply</i>	1	1.141378	1.579275
<i>sizeEffectOnDemand</i>	1.114942	1.069043	1.141962
<i>wealthToPopulationExponent</i>	0.776741	0.884321	0.463994
<i>distanceDecay</i>	0.672053	0.725789	0.710989
<i>populationToWealthExponent</i>	1.356863	1.023220	1
<i>bonusMultiplier</i>	0.082234	0	0.112958
<i>fixedCost</i>	0.000747	0	0.000949
<b>Rank-size slope</b>	-0.31	-2.54	-1.90
<b>Population difference</b>	$-33 \times 10^6$	$-18 \times 10^6$	$-41 \times 10^6$
<b>Inversion count</b>	1	2	1

doi:10.1371/journal.pone.0138212.t006

same value as an interurban exchange. The reduction of the model structure by muting two mechanisms could explain the variability of growth and mimics cities' localism observed empirically during the 1990s and 2000s in Russia. However, pattern H seems a lot more reasonable than F in terms of the continuity of the (de)hierarchisation trend.

In G, shrinkage is stronger for the largest cities while the smallest ones continue to grow. This configuration is obtained by making consumptive behaviour superlinear (*sizeEffectOnDemand* = 1.11) and productive behaviour linear (*sizeEffectOnSupply* = 1) with city size. The advantage given to urban aggregates of low population results in the loss of attractivity (and population) of cities above 100,000 inhabitants and the growth of cities below that point. This

configuration is quite unlikely in the present conditions, where agglomeration economies and integration in the global networks are decisive of city growth, but constitutes an attractor to be considered.

Finally, pattern I seems a possible pattern to be observed in the future. In this configuration, a couple of large cities maintain a growing evolution in an overall shrinking context. The smaller the city, the more it shrinks. This realistic result is obtained with quite unlikely values for parameters (scaling exponent of production  $sizeEffectOnSupply > 1.5$  and linear distribution of wealth with regards to population:  $populationToWealthExponent = 1$ ). Patterns G and I could represent (extreme) stylised scenarios to be explored if one was interested in simulating the evolution of the Former Soviet Union in a prospective manner, because they correspond to two regimes of relationship between cities and the economy: one in which large cities have a clear disadvantage in producing and exchanging, and one in which they are the economic winner of the population distribution. Those scenarios could represent the extent to which congestion costs over- or under-compensate for agglomeration economies in a system of cities experiencing shrinkage.

The analysis above should be seen as one step, unexpected patterns inspection, from the wider iterative process mentioned in the introduction. The methodology we propose is to start with exploring the model's behaviour as widely as possible, and to use the extreme patterns that are found to detect problems in the model's assumptions and revise them. This is why the parameters bounds in [Table 4](#) are very wide and may seem unrealistic. Rather than trying to guess some reasonable bounds a priori and run the exploration within them, we are trying the opposite approach: guess what bounds are reasonable based on what patterns we observe. The reason for this is that with complex systems, we may have data about the emerging patterns but not so much about the internal mechanisms, and this is the case with city systems (it is easier to know the population size in time through censuses than quantities related to MARIUS's parameters). The very purpose of complex systems modelling is sometimes to capture internal mechanisms that are not directly observable. In such case, it is easier to say if an observed global pattern is realistic rather than if a parameter value is. This experiment should be seen as just one step of the incremental work aiming at gradually revising the model's assumptions (both in the model's internal mechanisms and the parameter bounds) based on the unexpected patterns observed in simulation. Their interpretation above suggests at least that some of the parameters bounds should be revised and the exploration performed again. The continuation of this iterative process will be done in future work focusing on the analysis of the MARIUS model rather than on the PSE algorithm.

## Conclusion

Following Popper's argument about testing the validity of a scientific theory, we argued in the introduction that looking for unexpected patterns a model can produce, its potential falsifiers, is an important part of the modelling process usually left aside. It has two benefits: to give the modellers the opportunity to revise their model by confronting it to empirical knowledge, and to reveal the model's predictions.

To discover the unexpected patterns a model can produce, we presented an evolutionary method based on Novelty Search which explores the parameter space of a model looking for diversity of the output patterns, named Pattern Space Exploration or PSE. We tested the method on a model of the well studied phenomenon of collective motion, and then applied it to a model of system of cities, looking at the relationship between city hierarchisation and population growth.



During the tests on the collective behaviour model, the method successfully found the five main expected classes of behaviours: gas, moving liquid, moving solid, stationary liquid and stationary solid. The method's efficiency to discover a wide range of diverse patterns was compared to the a priori samplings LHS, Sobol, and a regular grid. The results of these comparisons are clearly in favour of the method we propose. It suggests that sampling the parameter space, even with methods that have good space filling properties such as LHS and Sobol's, is not a good strategy to explore the pattern space generated by a model. Driving the exploration directly in the pattern space, with a method like the one we propose here, is better suited.

The purpose of the experiment with the model MARIUS was to find out what kind of evolution of a system of cities the model could produce, in terms of city hierarchisation and population growth. We found that all combinations of hierarchisation and equalisation on the one hand, and population growth and shrinkage on the other, could be exhibited by the model. We also found that the model predicted values of hierarchisation and equalisation lying far beyond the intervals commonly found in census data. The discovery of unexpected patterns is the core contribution of the method. They can either be counter-examples or predictions of the model. A detailed study of how and why the model produces these patterns is an good opportunity to either revise the model, or make interesting predictions.

While motivated by model falsification to build robust models, this work focuses on the presentation of an algorithm to explore widely the pattern space of a model and on showing experimentally its efficiency in discovering various patterns. Such an exploration is key to discovering the model's unexpected patterns which serve as a basis for falsification. Future work with the model MARIUS will focus more on using PSE for the iterative process of model falsification and revision for robust model building.

One could think of several ways to achieve the search for diversity in the pattern space with an evolutionary algorithm. We experimented with the one which seemed the simplest: at the selection stage, select the individuals whose behaviours were the least encountered. Other alternatives come to mind. In particular, one could think of selecting the individuals which when used for replication through crossover and mutation produce offsprings with the most diverse behaviours. Such alternatives may constitute directions for future research.

The incentive for the development of PSE is to provide tools for good modelling practice and robust model development. We hope this paper will attract modellers' attention on an important flaw in today's common model validation method based on corroboration. Robust model design must also rely on falsification. The PSE method is a proposition to help modellers discover both potential flaws and predictions of their models.

## Supporting Information

### **S1 Appendix. MARIUS ODD description.**

(PDF)

**S1 Fig. Patterns discovered for each PSE replication.** Rows show the patterns discovered by each replicated run of PSE using the flocking model. See also caption of [Fig 4](#).

(PNG)

**S2 Fig. Patterns discovered for each LHS replication.** Idem, with the LHS a priori sampling.

(PNG)

## Acknowledgments

This work was founded by the ERC GeoDiverCity led by Denise Pumain. Results obtained in this paper were computed on the biomed and the vo.complex-system.eu virtual organization of

the European Grid Infrastructure (<http://www.egi.eu>). We thank the European Grid Infrastructure and its supporting National Grid Initiatives (France-Grilles in particular) for providing the technical support and infrastructure. We would also like to thank Forrest Stonedahl for in depth and useful comments on the manuscript.

## Author Contributions

Conceived and designed the experiments: GC CC RR. Performed the experiments: GC CC RR. Analyzed the data: GC CC RR. Contributed reagents/materials/analysis tools: GC CC RR. Wrote the paper: GC CC RR.

## References

1. Loeuille N. Influence of evolution on the stability of ecological communities. *Ecology letters*. 2010; 13(12):1536–1545. doi: [10.1111/j.1461-0248.2010.01545.x](https://doi.org/10.1111/j.1461-0248.2010.01545.x) PMID: [21054734](https://pubmed.ncbi.nlm.nih.gov/21054734/)
2. Goss S, Aron S, Deneubourg JL, Pasteels JM. Self-organized Shortcuts in the Argentine Ant. *Naturwissenschaften*. 1989; 76:579–581. doi: [10.1007/BF00462870](https://doi.org/10.1007/BF00462870)
3. Gerstner W, Sprekeler H, Deco G. Theory and simulation in neuroscience. *Science*. 2012; 338(6103):60–65. doi: [10.1126/science.1227356](https://doi.org/10.1126/science.1227356) PMID: [23042882](https://pubmed.ncbi.nlm.nih.gov/23042882/)
4. Schelling T. Sorting and Mixing: Race and Sex. In: *Micromotives and Macrobehavior*. W. W. Norton & Co.; 1978. p. 137–155.
5. Menaut JC, Gignoux J, Prado C, Clobert J. Tree Community Dynamics in a Humid Savanna of the Cote-d'Ivoire: Modelling the Effects of Fire and Competition with Grass and Neighbours. *Journal of Biogeography*. 1990; 17:471–481. doi: [10.2307/2845379](https://doi.org/10.2307/2845379)
6. Grégoire G, Chaté H, Tu Y. Moving and staying together without a leader. *Physica D: Nonlinear Phenomena*. 2003; 181(3):157–170. doi: [10.1016/S0167-2789\(03\)00102-7](https://doi.org/10.1016/S0167-2789(03)00102-7)
7. Grimm V, Railsback SF. Pattern-oriented modelling: a 'multi-scope' for predictive systems ecology. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 2012; 367(1586):298–310. doi: [10.1098/rstb.2011.0180](https://doi.org/10.1098/rstb.2011.0180)
8. Windrum P, Fagiolo G, Moneta A. Empirical Validation of Agent-Based Models: Alternatives and Prospects. *Journal of Artificial Societies and Social Simulation*. 2007; 10(2):8.
9. Moss S. Alternative Approaches to the Empirical Validation of Agent-Based Models. *Journal of Artificial Societies and Social Simulation*. 2008; 11(1):5. Available from: <http://jasss.soc.surrey.ac.uk/11/1/5.html>
10. Rossiter S, Noble J, Bell KRW. Social Simulations: Improving Interdisciplinary Understanding of Scientific Positioning and Validity. *Journal of Artificial Societies and Social Simulation*. 2010; 13(1):10.
11. Marchionni C, Ylikoski P. Generative Explanation and Individualism in Agent-Based Simulation. *Philosophy of the Social Sciences*. 2013; 43(3):323–340. doi: [10.1177/0048393113488873](https://doi.org/10.1177/0048393113488873)
12. Batty M, Torrens PM. Modeling complexity: the limits to prediction. *Cybergeo: European Journal of Geography*. 2001; doi: [10.4000/cybergeo.1035](https://doi.org/10.4000/cybergeo.1035)
13. Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, et al. *Experimental Designs. In: Global sensitivity analysis: the primer*. John Wiley & Sons; 2008. p. 53–107.
14. Sanchez SM, Wan H. Better than a petaflop: The power of efficient experimental design. In: *Winter Simulation Conference. Winter Simulation Conference*; 2009. p. 60–74.
15. Kleijnen JP, Sanchez SM, Lucas TW, Cioppa TM. State-of-the-art review: a user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing*. 2005; 17(3):263–289. doi: [10.1287/ijoc.1050.0136](https://doi.org/10.1287/ijoc.1050.0136)
16. Gramacy RB, Lee HK, Macready WG. Parameter space exploration with Gaussian process trees. In: *Proceedings of the twenty-first international conference on Machine learning*. ACM; 2004. p. 45.
17. Gramacy RB, Lee HK. Adaptive design and analysis of supercomputer experiments. *Technometrics*. 2009; 51(2):130–145. doi: [10.1198/TECH.2009.0015](https://doi.org/10.1198/TECH.2009.0015)
18. Castro P, Petit E, Farjallah A, Jalby W. Adaptive sampling for performance characterization of application kernels. *Concurrency and Computation: Practice and Experience*. 2013; 25(17):2345–2362. doi: [10.1002/cpe.3097](https://doi.org/10.1002/cpe.3097)

19. Schmitt C, Rey-Coyrehourcq S, Reuillon R, Pumain D. Half a billion simulations: Evolutionary algorithms and distributed computing for calibrating the SimpopLocal geographical model. *Environment and Planning B*. 2015; 42(2):300–315. doi: [10.1068/b130064p](https://doi.org/10.1068/b130064p)
20. Calvez B, Hutzler G. Parameter space exploration of agent-based models. In: *Knowledge-Based Intelligent Information and Engineering Systems*. Springer; 2005. p. 633–639.
21. Terano T. Exploring the vast parameter space of multi-agent based simulation. In: *Multi-agent-based simulation VII*. Springer; 2007. p. 1–14.
22. Stonedahl F, Wilensky U. Finding forms of flocking: Evolutionary search in abm parameter-spaces. In: *Multi-Agent-Based Simulation XI*. Springer; 2011. p. 61–75.
23. Miller JH. Active nonlinear tests (ANTs) of complex simulation models. *Management Science*. 1998; 44(6):820–830. doi: [10.1287/mnsc.44.6.820](https://doi.org/10.1287/mnsc.44.6.820)
24. Stonedahl F, Wilensky U. Evolutionary Robustness Checking in the Artificial Anasazi Model. In: *AAAI Fall Symposium: Complex Adaptive Systems*; 2010. p. 120–129.
25. Lehman J, Stanley KO. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*. 2011; 19(2):189–223. doi: [10.1162/EVCO\\_a\\_00025](https://doi.org/10.1162/EVCO_a_00025) PMID: [20868264](https://pubmed.ncbi.nlm.nih.gov/20868264/)
26. Lehman J, Stanley KO. Efficiently evolving programs through the search for novelty. In: *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM; 2010. p. 837–844.
27. Cottineau C, Chapron P, Reuillon R. Growing models from the bottom up. An evaluation-based incremental modelling method (EBIMM) applied to the simulation of systems of cities. *Journal of Artificial Societies and Social Simulation (JASSS)*. 2015; 18(8):In press.
28. Hinterding R. Gaussian mutation and self-adaption for numeric genetic algorithms. In: *Evolutionary Computation, 1995.*, IEEE International Conference on. vol. 1. IEEE; 1995. p. 384.
29. Eiben ÁE, Hinterding R, Michalewicz Z. Parameter control in evolutionary algorithms. *Evolutionary Computation*, IEEE Transactions on. 1999; 3(2):124–141. doi: [10.1109/4235.771166](https://doi.org/10.1109/4235.771166)
30. Deb K. An Efficient Constraint Handling Method for Genetic Algorithms. In: *Computer Methods in Applied Mechanics and Engineering*; 1998. p. 311–338.
31. Reuillon R, Leclaire M, Rey-Coyrehourcq S. OpenMOLE, a workflow engine specifically tailored for the distributed exploration of simulation models. *Future Generation Computer Systems*. 2013; 29(8):1981–1990. doi: [10.1016/j.future.2013.05.003](https://doi.org/10.1016/j.future.2013.05.003)
32. Czirók A, Stanley HE, Vicsek T. Spontaneously ordered motion of self-propelled particles. *Journal of Physics A: Mathematical and General*. 1997; 30(5):1375. doi: [10.1088/0305-4470/30/5/009](https://doi.org/10.1088/0305-4470/30/5/009)
33. Vicsek T, Czirók A, Ben-Jacob E, Cohen I, Shochet O. Novel type of phase transition in a system of self-driven particles. *Physical review letters*. 1995; 75(6):1226. doi: [10.1103/PhysRevLett.75.1226](https://doi.org/10.1103/PhysRevLett.75.1226) PMID: [10060237](https://pubmed.ncbi.nlm.nih.gov/10060237/)
34. Toner J, Tu Y, Ramaswamy S. Hydrodynamics and phases of flocks. *Annals of Physics*. 2005; 318(1):170–244. doi: [10.1016/j.aop.2005.04.011](https://doi.org/10.1016/j.aop.2005.04.011)
35. Reynolds CW. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*. 1987; 21(4):25–34. doi: [10.1145/37402.37406](https://doi.org/10.1145/37402.37406)
36. Wilensky U. NetLogo flocking model [Software]. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL; 1998. Available from: <http://ccl.northwestern.edu/netlogo/models/Flocking>
37. Wilensky U. NetLogo [Software]. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL; 1999. Available from: <http://ccl.northwestern.edu/netlogo/>
38. Byrne DS. *Complexity theory and the social sciences: an introduction*. Psychology Press; 1998.
39. Pumain D. Une approche de la complexité en géographie. *Geocarrefour*. 2003; 78(1):25–31. doi: [10.4000/geocarrefour.75](https://doi.org/10.4000/geocarrefour.75)
40. Warf B. Postmodernism and the localities debate: ontological questions and epistemological implications. *Tijdschrift voor Economische en Sociale Geografie*. 1993; 84(3):162–168. doi: [10.1111/j.1467-9663.1993.tb00646.x](https://doi.org/10.1111/j.1467-9663.1993.tb00646.x)
41. Axtell R, Axelrod R, Epstein JM, Cohen MD. Aligning simulation models: A case study and results. *Computational & Mathematical Organization Theory*. 1996; 1(2):123–141. doi: [10.1007/BF01299065](https://doi.org/10.1007/BF01299065)
42. Moriconi-Ebrard F. *L'Urbanisation du Monde: depuis 1950*. Anthropos, Collection Villes; 1993.
43. Slepukhina I. *Russian cities at the crossroads: getting lost in transition or moving towards regeneration [Ph.D. thesis]*. Politecnico di Milano, Italy; 2014.
44. Cottineau C. *L'évolution des villes dans l'espace post-soviétique Observation et modélisations [Ph.D. thesis]*. University Paris 1 Panthéon-Sorbonne, France; 2014.
45. Spulber DF. *Global competitive strategy*. Cambridge University Press Cambridge; 2007.

46. Combes PP, Lafourcade M. Revue de la littérature académique quantifiant les effets d'agglomération sur la productivité et l'emploi; 2012.
47. Ciccone A, Hall RE. Productivity and the density of economic activity. *The American Economic Review*. 1996; 86(1):54–70.
48. Bairoch P. De Jéricho à Mexico: villes et économie dans l'histoire. Gallimard Paris; 1985.
49. Pumain D, Swerts E, Cottineau C, Vacchiani-Marcuzzo C, Ignazzi A, Bretagnolle A, et al. Multilevel comparison of large urban systems. *Cybergeo: European Journal of Geography*. 2014; Available: <http://geodiversity.parisgeo.cnrs.fr/blog/resources/draft-papers/>
50. Soo KT. Zipf's Law for cities: a cross-country investigation. *Regional science and urban Economics*. 2005; 35(3):239–263. doi: [10.1016/j.regsciurbeco.2004.04.004](https://doi.org/10.1016/j.regsciurbeco.2004.04.004)