How Much Past to See the Future: A Computational Study in Calibrating Urban Cellular Automata

Ivan Blecic[a], Arnaldo Cecchini[a] and Giuseppe A. Trunfio[a]*

[a]Department of Architecture, Planning and Design, University of Sassari, P.zza Duomo, 6, 07041 Alghero (SS), Italy

(*Received 13 April 2014; final version received 21 September 2014*)

The objective of this computational study was to investigate to which extent the *availability* and the *way of use* of historical maps may affect the quality of the calibration process of Cellular Automata (CA) urban models. The numerical experiments are based on a constrained CA applied to a case study. Since the model depends on a large number of parameters, we optimize the CA using Cooperative Coevolutionary Particle Swarms, which is an approach known for its ability to operate effectively in search spaces with a high number of dimensions. To cope with the relevant computational cost related to the high number of CA simulations required by our study, we use a parallelized CA model that takes advantage of the computing power of Graphics Processing Units. The study has shown that the accuracy of simulations can be significantly influenced by both the number and position in time of the historical maps involved in the calibration.

Keywords: Cooperative Coevolution; GPGPU; Urban Models; Cellular Automata;

1. Introduction

A standing question haunts the application of Cellular Automata (CA) models for the simulation of urban dynamics and evolution of land uses in cities. Their capability to reproduce complex spatial patterns has long mesmerized urban scholars (White and Engelen 1993, Batty and Xie 1994, Clarke *et al.* 1997, Cecchini 1996) and fostered the development of notable CA-based applications, tools for land-use planning support, policy evaluation and future scenarios analysis (White and Engelen 2000, Barredo *et al.* 2003, Geertman and Stillwell 2004, Engelen *et al.* 2005, Lavalle *et al.* 2011, Blecic *et al.* 2014a). We observe a constant interest for such applications in the vogue of sustainable urban development, as tools providing decision-makers and urban planners information

*Corresponding author. Email: trunfio@uniss.it

about the scenarios of evolution of cities and urban regions, including the types, location and amount of change in land uses (Santé *et al.* 2010).

The question is though the one standing since the first steps of the pioneers of CA urban modelling: that of their empirical validity. While only time can ultimately have the last say if a model was effective (if perhaps not just lucky) to accurately predict the future, we should at least demand it to be able to describe the past, that is to say, to fit the known points in history. This in the operational use of CA models is the recurring issue of their *calibration*, to which this article intends to provide a contribution.

In CA modelling, a calibration procedure adapts the parameter-dependent transition rules in order to make the modelled urban phenomena fit the past data. Such estimation procedure of model parameters, for the purpose of improving the agreement between real and simulated phenomena, must be an essential phase of the model design process. Nevertheless, CA calibration is often a challenging problem, whose complexity grows remarkably with the number of parameters. It is well known that calibration through trial-and-errors based only on expert knowledge is time consuming, usually leads to unreliable results and is a hard-won wisdom. The current research trends instead suggest the use of formal, structured and automated optimization procedures (Avolio *et al.* 2006, Goldstein 2003, Feng *et al.* 2011, Rabbani *et al.* 2012, Li *et al.* 2013).

Such automatic calibrations crucially depend on the *availability* of historical maps of the area under study, which are used, together with an appropriate metric of agreement, to guide the search of the values of model parameters. Important questions, to which this article tries to provide some answers, concern the relationship between the predictive capability of the calibrated model and the historical maps used in the optimization process. For example, for a CA modeller it is relevant to have an idea of the possible calibration error, with the related loss of model accuracy, which may be expected when very few historical maps are available to support the calibration process. On the other hand, having a rich time-series of the past land-use patterns for the area under study may not bring benefits in all circumstances due to possible model overfitting.

In spite of the importance of these issues, to our knowledge, in the literature there are no systematic studies on the relation between the number and position in time of the exploited historical maps and the resulting calibration error. As a contribution towards overcoming this lack, the main objective of this study was to investigate, for a typical urban CA, to which extent the *availability* and the *way of use* of past maps may affect the calibration process and ultimately the predictive performance of the resulting model. The article provides several insights about what is to be expected in terms of accuracy of model calibration, under different usage scenarios of spatio-temporal data.

The discussion we present is grounded on a computational study of a constrained CA for simulating urban dynamics. To perform an empirical investigation based on the execution of several millions of CA simulations, we took advantage of a parallel variant of the constrained CA model, using general-purpose computing on graphics processing unit (GPGPU). In order to cope with the high number of parameters involved in the calibration, we used a state-of-the-art metaheuristic, the Cooperative Coevolutionary Particle Swarm optimization (CCPSO) (vandenBergh and Engelbrecht 2004), which is a variation of the standard Particle Swarm Optimization (PSO) algorithm (Kennedy and Eberhart 1995) specifically designed to deal with optimizations in high-dimensional search spaces.

The article is organized as follows: in section 2 we outline the GPU-accelerated CA model used in the numerical experiments. In section 3 we formalize the CA calibration problem and in section 4 we describe the CCPSO algorithm adopted for optimization.

Then, in section 5, we illustrate and discuss the numerical experiments. Finally, in section 6, we draw some conclusions and formulate some hypotheses for future work.

2. The Parallelized Constrained CA Urban Model

We use a CA, representing the geographical space of interest, which evolves through discrete time steps in order to simulate a real land-use dynamics over time.

The relevant component of the state of each cell is its land-use class (such as *residential*, *industrial*, *commercial* and *agriculture*). Cells may also hold other relevant information, such as their distance from the main transportation networks, constraints related to zoning regulations and cells' physical features (slope, elevation, etc.). During the simulation, each cell can change its land use depending on its neighbouring cells and its internal state. However, the cell's state transition also depends on global constraints on the land of each land-use type to be assigned at each time step.

We define two main categories of land uses: *dynamic*, which can change during the simulation, and *static*, which do not change, but may influence dynamic uses by exerting attractive or repulsive effects on them (White and Engelen 2000, Engelen *et al.* 2005). Typical static land uses may represent transportation network, public services and facilities, and so on. The category of dynamic uses can be further split into *active* uses, for which there is an explicit demand at each CA step, and *passive* uses, representing land available to be transformed into active uses.

The local CA dynamics is determined by the so-called *transition potentials* P_j, which is a very common way to express a cell's propensity to acquire the j-th type of use (Wu 1998, Cheng and Masser 2004, He *et al.* 2008, Blecic *et al.* 2004). At the beginning of each CA step, the values P_j are computed for each cell and each active land use through the following equation:

$$P_j = I_i + \gamma_j \, S_j \, Z_j \, N_j \tag{1}$$

where:

- i is the current cell's land use and $I_i \geq 0$ represents an inertia due to the transition costs from the use i to a different use;
- $\gamma_j = 1 + (-\ln \psi)^{\alpha_j}$, where ψ is a random number between 0 and 1 and α_j is a parameter that defines the degree of randomness;
- $S_j \in [0, 1]$ is the so called *suitability* factor for the active land use j. The suitability represents the propensity of a cell to sustain a particular activity or land use. In this study, the values S_j are expressed as a logistic function of n_f local predictors x_k (e.g. the cell's distance from the street network or the terrain slope) as follows:

$$S_j = \frac{exp(\sum_{k=1}^{n_f} b_{kj} x_k)}{1 + exp(\sum_{k=1}^{n_f} b_{kj} x_k)} \tag{2}$$

  where the parameters b_{kj} are estimated through the calibration process.
- $Z_j \in [0, 1]$ defines the degree of legal or planning permissibility of the j-th land use (for example due to zoning regulations by the planning authority);
- N_j is the so called *neighbourhood effect*, which represents the sum of all the attractive and repulsive effects of land uses within the neighbourhood on the j-th use which the cell under consideration may assume. In particular, the factor N_j is computed as:

$$N_j = \nu + \sum_{c \in V} \phi_{i,j}(\delta_c) \tag{3}$$

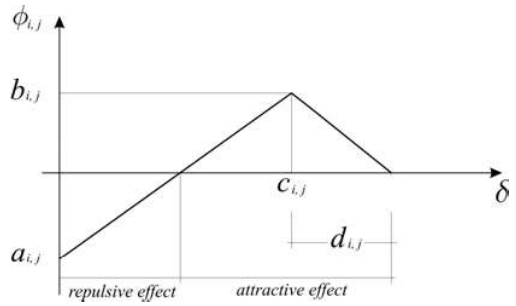

Figure 1. The function $\phi_{i,j}(\delta)$ represents the influence of the i-th land use at the distance δ on the potential land use j.

where the summation is extended over all the cells of the cell's neighbourhood V and: i denotes the current land use of the cell $c \in V$, δ_c is the distance from the neighbouring cell c, and $\phi_{i,j}(\delta)$ is a parameterized function expressing the influence of the i-th land-use at the distance δ on the potential land use j. The term ν is a constant value computed before the beginning of the simulation so that $N_j \geq 0$.

As shown in Fig. 1, we assume that the $\phi_{i,j}(\delta)$ are piecewise linear functions depending on four scalar parameters: $a_{i,j}$ and $b_{i,j}$ are defined in the interval $[-\sigma, \sigma]$, where $\sigma \in \mathbb{R}^+$ is a suitable positive constant; $c_{i,j}$ and $d_{i,j}$ are defined in the interval $[0, \delta_{\max}]$, where $\delta_{\max} \in \mathbb{R}^+$ represents the maximum distance of influence of a land use onto another. This choice allows to model a wide variety of relevant situations in which the influence generated by an urban function depends on the distance.

As reported by (Santé *et al.* 2010), different types of neighbourhoods have been used in the literature for simulating urban dynamics through CA, including square and circular regions of different sizes. In the model adopted here, the CA neighbourhood V is defined as the square region having a sufficient size to allow local-scale spatial processes to be captured by the CA transition rules. Compared with a circular neighbourhood, a square region allows an easier management and a greater computational efficiency in our parallel implementation on the GPU. Also, we verified that the issue of potential distortions, as reported by (Li and Yeh 2000), in our case is not significant. In fact, we use a rather large neighbourhood (i.e. a square with the side of 40 cells) and the functions $\phi_{i,j}$ used in practice are rapidly decreasing with distance.

Once the cells propensities P_j to assume alternative land uses have been calculated, the land use in the next step of the CA evolution must be computed for each cell. As in most CA-based urban models (White and Engelen 1993, White *et al.* 1997, Wu 1998, Cheng and Masser 2004, Blecic *et al.* 2004), the total land area that changes from its current land use to a different use is determined by an external constraint. This implies that the state-transition phase must take place on a non-local basis. Typically, this phase consists in transforming each cell into the state with the highest potential, given the exogenous constraint on the overall number of cells in each state imposed for that step (White and Engelen 2000, Engelen *et al.* 2005).

2.1. *GPGPU implementation*

Following the GPGPU approach proposed in (Blecic *et al.* 2013), we developed a parallel version of the CA model described above. In particular we used the GPUs provided by nVidia, consisting of a group of Streaming Multiprocessors (SMs) capable to support a significant number of co-resident concurrent threads. Each SM on its turn consists of multiple Scalar Processor (SP) cores. To develop the model, we have adopted the popular C-language Compute Unified Device Architecture (CUDA).

In order to obtain high efficiency, we have formulated the parallelization to avoid significant memory transfers between the CPU and GPU during the simulation.

In the model, the main phases of each CA step are: (*i*) the computation of the transition potentials following Eqs. (1) and (3); (*ii*) the constrained assignment of the future land uses. The first phase can be carried out independently for each cell. Therefore, it is well suited to be parallelized according to the computational model offered by GPUs, just associating a thread to each cell of the CA. Instead, for the second phase we adopted the ad-hoc algorithm of constrained state transitions proposed in (Blecic *et al.* 2013). In fact, a very poor speedup would be achieved through the simple translation in CUDA of the constrained procedure used in (White and Engelen 2000, Engelen *et al.* 2005), which is inherently sequential.

Using the recent CUDA device nVidia Tesla K40, endowed with 2880 thread processors, the adopted GPGPU approach leads to speedups that can easily exceed 100 times the execution time of the corresponding sequential run on a standard workstation. This was a key factor to allow our computational study. Obviously, in addition to the used hardware, the actual acceleration depends on many other factors, including the number of cells of the CA, the neighbourhood radius as well as the number of involved land uses. More details on the GPU-accelerated CA model can be found in (Blecic *et al.* 2013).

3. Automatic Calibration

The dynamics of the CA model described above depends on many scalar parameters that must be adapted to the specific application context. In particular, if the model includes n_t land uses, n_d of which dynamic and n_a actively modelled, it depends on:

- n_a parameters I_k defining the inertial contribution to the transition potentials;
- $n_f\, n_a$ parameters involved in the logistic suitability defined by Eq. (2);
- $4\, n_t\, n_a$ parameters a_{ij}, b_{ij}, c_{ij}, d_{ij} involved in the piecewise functions $\phi_{i,j}$ of Eq. (3);
- n_a parameters α_j defining the degree of randomness.

All the above parameters can be collected in a vector $\mathbf{p}$ belonging to a D-dimensional search space Λ, with $D = n_a + n_f\, n_a + 4\, n_t\, n_a + n_a$.

With respect to $\mathbf{p}$, the model can be optimized to maximize the fitting between the simulated patterns and the real ones. To formalize the problem, let us suppose the existence of a spatio-temporal *training set* $\bar{\mathcal{V}}$ collecting a sequence of land-use maps of the area under study. In particular, let $\bar{\mathcal{V}}$ be composed of a series of q maps:

$$\bar{\mathcal{V}} = \left\{ \bar{\omega}^{(t)} : t \in \{\tau_1, \ldots, \tau_q\} \right\} \tag{4}$$

where the attribute τ_i indicates the time step in which the configuration $\bar{\omega}^{(t)}$ is known. For example, assuming that a CA step corresponds to one year, $\omega^{(0)}$ would represent a map of land uses of the area under study at the year t_0 and $\bar{\omega}^{(t)}$ would be the known land use configuration after t years.

Therefore, starting from the known configuration $\omega^{(0)}$, and given a vector $\mathbf{p}$ of parameters, the CA can be executed for the computation of the q automaton configurations:

$$\mathcal{V} = \left\{ \omega^{(t)} : t \in \{\tau_1, \ldots, \tau_q\} \right\} \tag{5}$$

where each $\omega^{(t)}$ is given by the CA simulation. The agreement between the real spatio-temporal sequence and the simulated one should be quantified through a suitable mea-

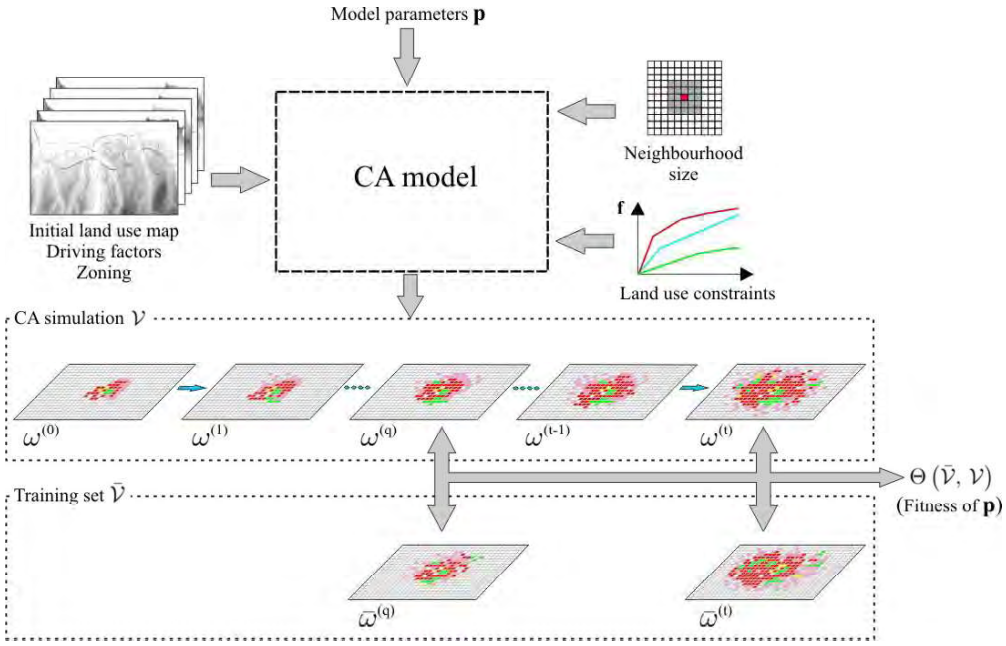

Figure 2. Scheme of the procedure for computing the fitness $\Theta(\bar{\mathcal{V}}, \mathcal{V})$ associated to a vector of parameters **p**.

sures of fitness $\Theta(\bar{\mathcal{V}}, \mathcal{V})$, which is computed for each value of **p** as shown in Fig. 2.

Thus, the automatic calibration process consists of using a suitable search algorithm to find a value of $\mathbf{p} \in \Lambda$ that maximize $\Theta(\bar{\mathcal{V}}, \mathcal{V})$. It is worth to point out that in general, as already shown in (Straatman *et al.* 2004), CA calibration procedures do not, in the line of principle, yield a unique solution. In other words, depending upon the structure of both the CA and the fitness function, the same CA configuration could be obtained using different parameter vectors.

In this study, we have based the fitness $\Theta(\bar{\mathcal{V}}, \mathcal{V})$ on the so called *Kappa simulation* K_s (van Vliet *et al.* 2011), which is a modified version of the standard Kappa statistic (Cohen 1960). In this indicator, the agreement between a 'real' map $\bar{\omega}$ and the corresponding simulated map ω is corrected to account for the amount of land-use transitions, which are computed taking as a reference the initial map $\omega^{(0)}$. To explain in detail the computation of K_s, in the following we indicate with $p(\bar{\omega} = i \wedge \omega = i)$ the fraction of cells having the same land use i in both maps $\bar{\omega}$ and ω. Also, we indicate with $p(\omega^{(0)} = i)$ the fraction of cells having the land use i in the initial map $\omega^{(0)}$ and with $p(\omega = i \mid \omega^{(0)} = j)$ the fraction of cells that changed from land use j in the initial map to land-use i in the map ω. According to (van Vliet *et al.* 2011), the value of K_s is defined as:

$$K_s = \frac{p_o - p_e}{1 - p_e} \tag{6}$$

where p_o is the observed fraction of agreement:

$$p_o = \sum_{i=1}^{n_d} p(\bar{\omega} = i \wedge \omega = i)$$

and p_e is the expected fraction of agreement (i.e. the agreement by chance), given the sizes of land-use class transitions:

$$p_e = \sum_{j=1}^{n_d} p(\omega^{(0)} = j) \cdot \sum_{i=1}^{n_d} p(\bar{\omega} = i \mid \omega^{(0)} = j) \cdot p(\omega = i \mid \omega^{(0)} = j)$$

As in the standard Kappa, a K_s value greater than 0 indicates that there is more similarity than random and a value of 1 indicates the perfect agreement between the two maps.

Using Eq. (6) and given a training set $\bar{\mathcal{V}}$ and a simulated set $\mathcal{V}$, the value of the fitness $\Theta\left(\bar{\mathcal{V}}, \mathcal{V}\right)$ is computed as follows:

- first, we determine the agreements $K_s(\bar{\omega}^{(t)}, \omega^{(t)})$ between all the couples of corresponding maps $\omega^{(t)} \in \mathcal{V}$ and $\bar{\omega}^{(t)} \in \bar{\mathcal{V}}$;
- then, we compute $\Theta\left(\bar{\mathcal{V}}, \mathcal{V}\right)$ as the arithmetic mean of all the $K_s(\bar{\omega}^{(t)}, \omega^{(t)})$.

For example, if the training set is composed of three historical maps corresponding to the CA steps 5, 10 and 15:

$$\bar{\mathcal{V}} = \left\{\bar{\omega}^{(5)}, \bar{\omega}^{(10)}, \bar{\omega}^{(15)}\right\}$$

then: (*i*) we execute the CA for computing the simulated configurations $\omega^{(5)}$, $\omega^{(10)}$ and $\omega^{(15)}$; (*ii*) using Eq. (6), we determine the agreements $K_s(\bar{\omega}^{(t)}, \omega^{(t)})$, for $t \in \{5, 10, 15\}$; (*iii*) finally, we compute $\Theta\left(\bar{\mathcal{V}}, \mathcal{V}\right)$ as:

$$\Theta\left(\bar{\mathcal{V}}, \mathcal{V}\right) = \frac{1}{3}\left[K_s(\bar{\omega}^{(5)}, \omega^{(5)}) + K_s(\bar{\omega}^{(10)}, \omega^{(10)}) + K_s(\bar{\omega}^{(15)}, \omega^{(15)})\right]$$

4. The Optimization Algorithm

Many optimization algorithms have been used for the automatic calibration of CAs (Clarke *et al.* 1997, Goldstein 2003, Avolio *et al.* 2006, Feng *et al.* 2011, Rabbani *et al.* 2012, Li *et al.* 2013, Liao *et al.* 2014, Blecic *et al.* 2010), including techniques based on exhaustive search, ad-hoc designed search procedures, as well as more general optimization metaheuristics.

For example, an exhaustive search in the parameter space has been used for SLEUTH (Clarke *et al.* 1997), which depends on only five parameters belonging to small domains of definition. Clearly, in case of models depending on a high number of real variables exhaustive search is impractical.

In other cases discussed in the literature, ad-hoc calibration techniques have been designed for specific urban models. For example, in (Straatman *et al.* 2004) the authors have developed an automatic calibration procedure specifically designed for a constrained CA. Their method involves an empirical search technique in which the *erroneous* neighbourhoods are first identified and then the parameters are adjusted to reduce the error. However, the method assumes that there exists a set of maps available on a year-by-year basis, which is rarely the case in practice. In (Engelen and White 2008) a further development of that method has been discussed to cope with the unavailability of maps covering each step of the calibration interval.

Another typical approach to automatic calibration consists of using optimization metaheuristics, such as, for example, various kind of Evolutionary Algorithms (EAs) (Goldstein 2003, Avolio *et al.* 2006, Feng *et al.* 2011, Rabbani *et al.* 2012, Li *et al.* 2013, Blecic *et al.* 2010). Compared to ad-hoc methods, the latter approach offers the potential advantage of being general and applicable to a larger class of models.

To tackle the optimization problem, in this study we use a variation of the standard PSO algorithm (Kennedy and Eberhart 1995), which was specifically designed to deal with optimizations in spaces with a high number of dimensions. The application of PSO to the calibration of urban models is not new (Feng *et al.* 2011, Rabbani *et al.* 2012, Liao *et al.* 2014). However, the examples discussed in the literature relate to cases with few

parameters, tackled with the use of fairly standard PSO. Instead, in this study we face the more challenging problem of calibrating a CA that depends on more than a hundred parameters. In the following we explain in details the adopted optimization approach.

4.1. *Cooperative Coevolutionary Particle Swarm Optimization*

PSO is a stochastic optimization metaheuristic (Kennedy and Eberhart 1995) in which a population (i.e. swarm) of *particles* is evolved step-by-step through the search space following rules inspired by the behaviour of flocks of birds. In PSO, particles change their position and velocity guided by the results obtained so far in terms of the fitness function value. In particular, the movements of a particle are influenced by the best position attained by that particle and by other current best-known positions in the search space. A particle's position and velocity at the step s of the search process is be computed using the following equations:

$$\mathbf{v}^{(s)} = w\,\mathbf{v}^{(s-1)} + c_1\,r_1\,\left[\mathbf{p}_{best}^{(s-1)} - \mathbf{p}^{(s-1)}\right] + c_2\,r_2\,\left[\mathbf{p}_{gbest}^{(s-1)} - \mathbf{p}^{(s-1)}\right] \tag{7}$$

$$\mathbf{p}^{(s)} = \mathbf{p}^{(s-1)} + \mathbf{v}^{(s)} \tag{8}$$

where $\mathbf{v}$ is the velocity of the particle, c_1 and c_2 are two positive constants, r_1 and r_2 are two random numbers uniformly drawn between 0 and 1, w is the so-called *inertia weight*, $\mathbf{p}^{(s)}$ is the position of the particle at step s, $\mathbf{p}_{best}^{(s-1)}$ is the best position reached by the particle up to step $s-1$ and $\mathbf{p}_{gbest}^{(s-1)}$ is the best-fitness point found so far by the whole swarm.

In this article we use CCPSO, which is a multipopulation PSO variant. In (Blecic *et al.* 2014b) it is shown that CCPSO can be much more effective than the standard PSO in optimizing a CA urban model with a high number of parameters.

In the CCPSO approach (Potter and De Jong 1994, vandenBergh and Engelbrecht 2004, Trunfio 2014) the original D-dimensional search space Λ is decomposed into K subspaces $\Lambda^{(i)}$ of the same dimension $D_K = D/K$:

$$\Lambda = \Lambda^{(1)} \times \Lambda^{(2)} \times \cdots \times \Lambda^{(K)} \tag{9}$$

Subsequently, each subspace $\Lambda^{(i)}$ is assigned to a different sub-population of n_K particles which operate as in the standard PSO. However, each individual carries only a portion of the candidate solution (i.e. some components of the solution vector). Its evaluation must hence be done obtaining the missing components from the proper sub-populations.

The approach proposed in (Potter and De Jong 1994), and adopted in this study, consists of combining the contributions of each sub-population in the so-called *context vector* $\mathbf{b}$. The latter collects the best individuals of each sub-population, providing in that way the missing vector components for evaluating the objective function.

In particular, let $\mathbf{y}^{(i)}$ be the D_K-dimensional vector representing the contribution of the i-th sub-population (i.e., its current best position in the subspace $\Lambda^{(i)}$):

$$\mathbf{y}^{(i)} = (y_1^{(j)}, y_2^{(j)}, \ldots, y_{D_K}^{(j)})^T$$

Then, the context vector is defined as:

$$\mathbf{y} = (\underbrace{y_1^{(1)}, \ldots, y_{D_K}^{(1)}}_{\mathbf{y}^{(1)}}, \underbrace{y_1^{(2)}, \ldots, y_{D_K}^{(2)}}_{\mathbf{y}^{(2)}}, \ldots, \underbrace{y_1^{(K)}, \ldots, y_{D_K}^{(K)}}_{\mathbf{y}^{(K)}})^T$$

Given the j-th particle $\mathbf{p}^{(i,j)} \in \Lambda^{(i)}$ of the i-th sub-population:

$$\mathbf{p}^{(i,j)} = (p_1^{(i,j)}, p_2^{(i,j)}, \ldots, p_{D_K}^{(i,j)})^T$$

its fitness value is obtained by running the CA with the vector of parameters $\mathbf{y}^{(i,j)} \in \Lambda$ defined as:

$$\mathbf{y}^{(i,j)} = (\underbrace{y_1^{(1)}, \ldots, y_{D_K}^{(1)}}_{\mathbf{y}^{(1)}}, \ldots, \underbrace{p_1^{(i,j)}, \ldots, p_{D_K}^{(i,j)}}_{\mathbf{p}^{(i,j)}}, \ldots, \underbrace{y_1^{(K)}, \ldots, y_{D_K}^{(K)}}_{\mathbf{y}^{(K)}})^T$$

In other words, in order to evaluate the fitness of $\mathbf{p}^{(i,j)}$ we construct the vector obtained from $\mathbf{y}$ by substituting the components provided by the i-th sub-population with the corresponding components of $\mathbf{p}^{(i,j)}$.

Except for this way of evaluating the individuals, the algorithm proceeds as the standard PSO in each subspace. However, when the i-th sub-population finds a new optimal point $\mathbf{y}^{(i)}$ in its subspace $\Lambda^{(i)}$, the context vector is updated accordingly.

Since most CA parameters are interdependent (i.e. their influence on the fitness depends on the value of other parameters), a problem may arise with the simple decomposition approach described above. In particular, when two interdependent CA parameters are assigned to different sub-populations the CCPSO algorithm exhibits a low search efficiency (Potter and De Jong 1994). For this reason, instead of using the static decomposition described above, in which the i-th sub-population operates on the subspace $\Lambda^{(i)}$ including all the directions in the interval $[(i-1) \times D_K + 1, \ldots, i \times D_K]$, we adopted the so-called *random grouping* (RG) strategy (Yang *et al.* 2008). In the latter, the subspaces $\Lambda^{(i)}$ are rebuilt at each iteration of the algorithm by assigning to each $d_k = D/K$ directions randomly selected without replacement from the set $\{1, 2, \ldots, D\}$. Such an approach increases the probability of having two interacting CA parameters in the same sub-population at least for some iteration of the search algorithm (Yang *et al.* 2008, Omidvar *et al.* 2010).

5. A Computational Study on CA Calibration

In this section we use the CA modelling approach and the computational techniques described above to empirically investigate different uses of historical data for model calibration. We first illustrate the adopted experimental setup, including the area under study, the detailed specifications of the model, the assumptions behind the numerical investigation and the used hardware. Then we present and discuss several computational experiments, highlighting some insights that may be derived for improving current calibration practices.

5.1. *Experimental set-up*

We applied the model to the area of the city of Heraklion, Crete. The CA representing the urban area was composed of 277×151 cells each with the side of 50 meters. We have used the urbanization map of the year 1980 shown in Fig. 3 to initialize the configuration $\omega^{(0)}$. Furthermore, we have assumed a time horizon of 50 years for the simulations of urban dynamics.

In the model we included $n_t = 10$ land-use types, $n_a = 4$ of which actively modelled: *residential dense*, *residential sparse*, *industrial areas*, and *commercial areas*. Passive land

uses are: *undeveloped land* (representing agricultural and natural land cover classes), and the *abandoned* state. Finally, static land uses are: *green urban areas and facilities*, *port area*, *airport area*, *water bodies*.

Moreover, we used the $n_f = 6$ driving factors mapped in Fig. 4 to determine the suitabilities given by Eq. (2), namely: *distance from highway*, *distance from main roads*, *terrain slope*, *altitude*, *distance from the sea* and *distance from the city center*. As suggested in (Liao *et al.* 2014) we normalized in the interval [0, 1] the driving factors as follows:

$$\overline{x}_i = \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}} \tag{10}$$

where $\overline{x}_i$ is the value used in Eq. (2) for the i-th factor, x_i is the original value, $x_i^{\min}$ and $x_i^{\max}$ are the minimum and maximum values, respectively.

To determine the influence of neighbouring land uses on the current cell during the simulations according to Eq. (3), we used for the square neighbourhood the side of 40 cells (i.e. 20 cells from the central cell).

One of the relevant design choices in urban CA is the number of discrete time step to simulate a given interval of time. As discussed in (Yeh and Li 2006), too few time steps cannot allow spatial interactions to generate suitable dynamics. On the other hand, too many time steps may imply high computational costs. In general, the optimal number of time steps is strictly related to the size of the CA cell and to the model parameters. However, within certain limits, when the choice of the number of steps is followed by a calibration of parameters, the latter are adapted accordingly. In this study, after some

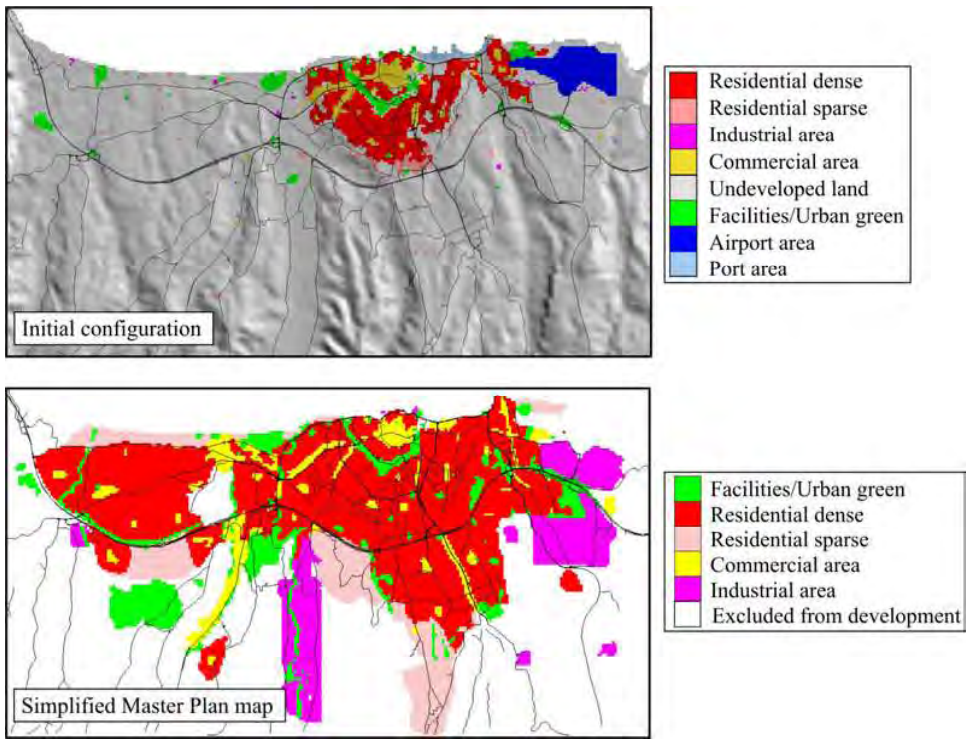

Figure 3. The initial and final CA configurations used for the calibration test. Note that Airport and Port areas were not included among the modelled land uses.

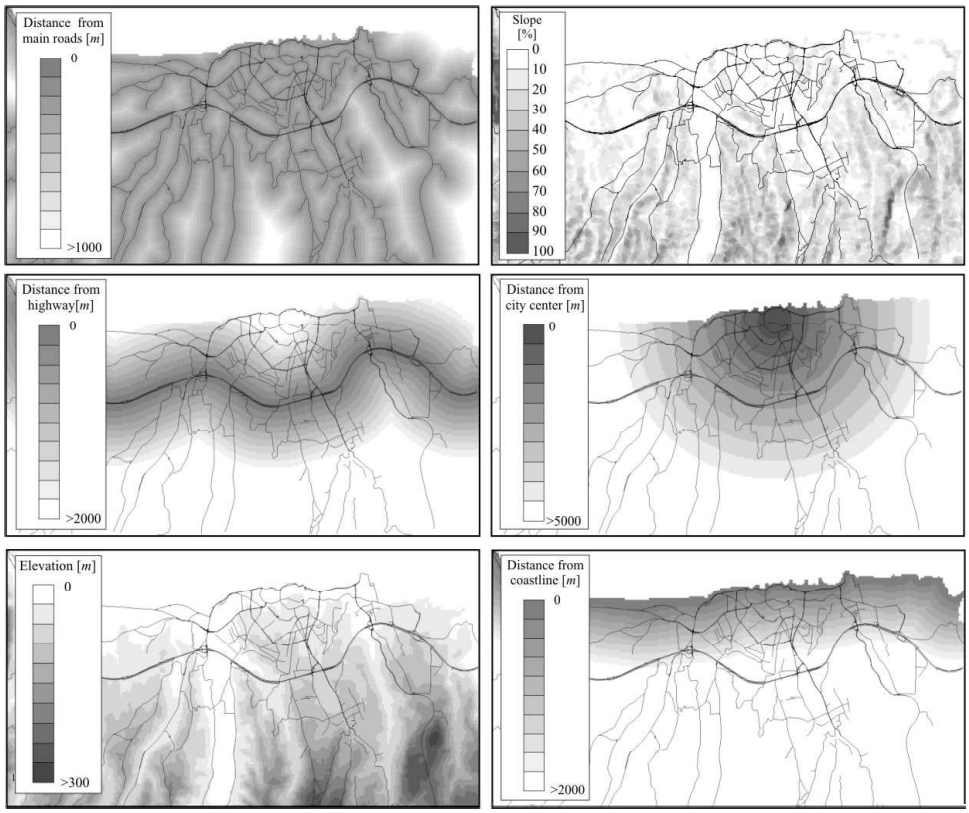

Figure 4. The maps of driving factors included in the model.

preliminary calibration experiments, we have chosen to use 25 CA steps for the adopted time horizon of 50 years.

As for the land use constraints, we adopted the scenario illustrated in Fig. 5. In practice, for the simulation step 15 we assumed the actual amount of land uses derived from a 2010 map of the area, with a linear variation between steps 0 and 15. Between steps 15 and 25 we assumed a hypothetical scenario of linear increment in demand for the active land uses.

In order to reduce the number of unknown model parameters, we neglected the influence exerted by *port area*, *airport area* and *abandoned areas* on their neighbouring cells. In addition, we adopted the value of 0.01 for all the α_i that define the simulation randomness. During the optimization we used a deterministic version of the model based on the same random seed for each simulation for the purpose of investigating the characteristics of calibrated models avoiding unessential complications related to the presence of a noisy objective function. Given the above specifications, the model depends on $D = 140$ unknown parameters.

Our computational study aims to investigate the impact of the used training set on the calibration error and, ultimately, on the accuracy of simulations. To this end, we use artificial training sets composed of maps generated through the CA itself. This, in comparison with the use of real historical land-use maps, has two advantages. First, the artificial training sets can be of the desired size (in the following, we use training sets extracted from a set $\bar{\mathcal{V}}$ of 25 maps that would be difficult to obtain in the reality). Second, this allows to exclude both the model error and errors related to data uncertainties, thus

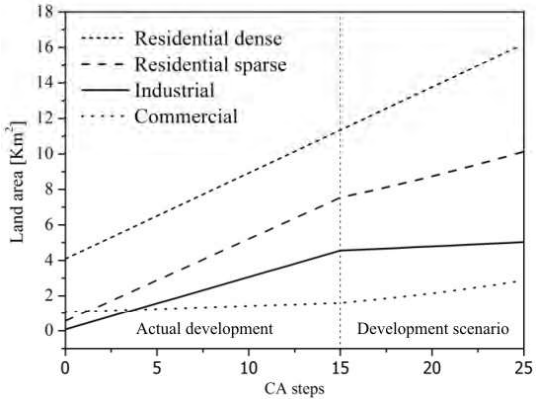

Figure 5. The scenario of land demand for the four actively modelled uses in the calibration tests.

Table 1. The drawn random parameters b_{kj} (see Eq. (2)) defining the influence of the following driving factors: *distance from highway* ($\bar{x}_1$), *distance from main roads* ($\bar{x}_2$), *terrain slope* ($\bar{x}_3$), *altitude* ($\bar{x}_4$), *distance from the sea* ($\bar{x}_5$) and *distance from the city center* ($\bar{x}_6$). The actively modelled land uses are: residential dense (RD), residential sparse (RS), commercial area (C), industrial area (I).

	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{x}_4$	$\bar{x}_5$	$\bar{x}_6$
RD	−0.726	0.343	−4.980	−0.021	−0.012	1.620
RS	0.121	0.338	−2.310	−0.033	0.712	0.234
I	0.426	0.643	−5.210	−0.043	−0.132	−0.237
C	0.335	0.214	−5.870	−0.032	−0.212	−0.228

Table 2. The drawn random parameters a, b, c, d (see Fig. 1) defining the mutual influence between the following land uses: residential dense (RD), residential sparse (RS), commercial area (C), industrial area (I). In bold are the influencing land uses.

		RD	RS	I	C
RD	a	202.4	−10.3	−12.7	17.9
	b	30.6	8.8	−2.6	25.2
	c	360.3	812.3	800.2	512.7
	d	598.3	187.2	167.3	475.4

		RD	RS	I	C
RS	a	98.4	−104.7	−18.2	13.4
	b	11.9	98.4	−3.3	18.7
	c	489.0	504.3	798.2	612.5
	d	498.7	491.2	194.2	375.4

		RD	RS	I	C
C	a	41.2	24.3	2.1	310.2
	b	66.7	36.7	1.2	50.2
	c	436.0	610.2	520.7	465.0
	d	562.6	385.3	477.3	521.2

		RD	RS	I	C
I	a	−32.6	−42.2	302.7	−6.2
	b	−41.9	7.8	4.3	15.3
	c	237.1	834.0	810.1	783.2
	d	726.4	102.0	183.2	211.7

showing more clearly the effects of calibration error.

To produce the artificial training sets, we randomly generated the 140 unknown parameters of the model within suitable ranges, trying to achieve a plausible land-use dynamics. Some of the drawn random parameters are listed in Tables 1 and 2.

Using such vector of random parameters and starting from the 1980 configuration, the CA simulation was then performed for 25 steps obtaining the corresponding set $\bar{\mathcal{V}}$ of 25 maps, partly shown in Fig. 6. The computational experiments discussed in the following essentially consist of using training sets $\bar{\mathcal{V}}_k \subset \bar{\mathcal{V}}$ with the aim of reproducing, with the best possible fit, all the configurations of $\bar{\mathcal{V}}$ after model calibration.

As for the CCPSO setup, we partitioned the search space into $K = 14$ groups of 10

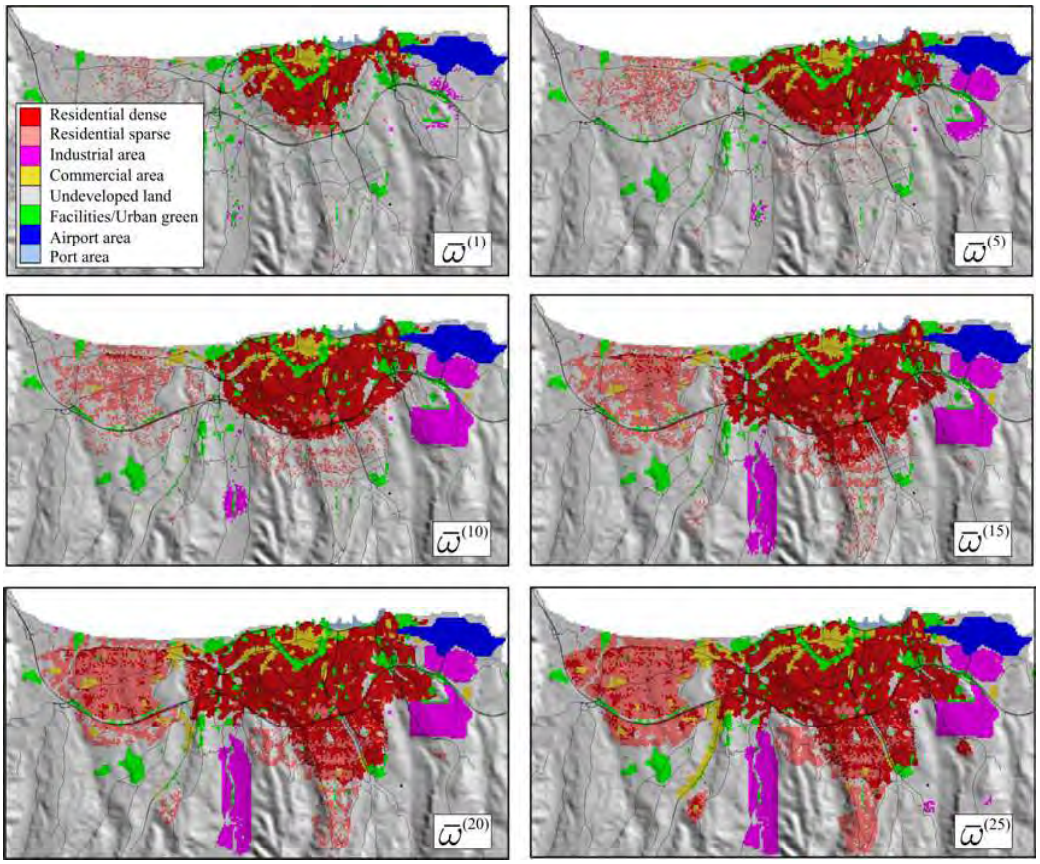

Figure 6. Some CA configurations produced through the randomly drawn model parameters and used in the calibration study.

parameters, applying random grouping every 4 iterations of the algorithm. To each group we assigned a swarm of 10 particles. For the PSO parameters, we adopted the following values suggested in (Kennedy and Clerc 2006): $w = 0.729844$ and $c_1 = c_2 = 1.49618$.

In order to ensure convergence, we assigned to each calibration a budget of 20000 CA evaluations, which proved sufficient according to the results.

In addition, it is well known that in optimizations based on probabilistic procedures the result might depend on the particular initialization of the population. Therefore, we carried out 20 independent runs for each training set $\bar{\mathcal{V}}_k$, averaging the results in terms of achieved fitness. Note that, in order to investigate the effect of different training sets, the i-th calibrations, one for each $\bar{\mathcal{V}}_k$, are initialized by the same seed while i-th and j-th $(i \neq j)$ calibrations have different initial seeds.

We ran the optimization algorithms on a workstation based on a Intel Xeon X5660 (2.80 GHz) and equipped with two different GPUs: the nVidia Tesla K40 and a nVidia Geforce GTX 680 graphic card. Both GPUs belong to the recent nVidia's Kepler GPU architecture. In order to exploit both GPUs, we developed a multi-GPU program using the C++/CUDA languages and a multi-threads approach. In particular, we organized the optimization process according to a master-slaves paradigm, in which the CPU executes the CCPSO algorithm while the two GPUs simultaneously carry out the CA simulations required for evaluating the particles.

5.2. *Results and discussion*

We organized the analysis of experiments in four parts. First, we investigate the achieved fitness through calibrations based on different training sets. Then, we use a statistical test to compare the accuracies of the simulations based on the previous calibrations. Subsequently, using the best set of parameters we analyse how the accuracy varies during the simulations. Finally, we compare the results in a forecasting perspective, that is, with the objective of predicting the correct final CA configuration, we investigate which combination of historical maps led to the best result.

5.2.1. Calibrations

We considered 15 different training sets composed of one or more maps taken from the set $\bar{\mathcal{V}}$. In the following, we indicate with $\bar{\mathcal{V}}_i$ a training set including only the map $\bar{\omega}^{(i)}$, with $\bar{\mathcal{V}}_{i,j}$ a training set including the maps $\bar{\omega}^{(i)}$, $\bar{\omega}^{(j)}$ and so forth. The statistics on the attained fitness are shown in Table 3. Also, in Fig. 7 we show some of the averaged convergence plots.

As expected, a calibration point just after the first CA step (i.e. training set $\bar{\mathcal{V}}_1$) gave rise to a relatively easy optimization problem. This is due to the very small differences between the two configurations $\omega^{(0)}$ and $\bar{\omega}^{(1)}$. Indeed, as can be seen in Fig. 7 the speed

Table 3. Statistics on the achieved fitness Θ over 20 independent runs in the calibration tests. The Table also shows the average time required by a single optimization.

Training set	Average	Best	Worst	Std. Dev.	Average time [s]
$\bar{\mathcal{V}}_1$	0.993	0.996	0.990	0.002	452
$\bar{\mathcal{V}}_5$	0.976	0.983	0.967	0.006	1121
$\bar{\mathcal{V}}_{10}$	0.968	0.977	0.959	0.006	2012
$\bar{\mathcal{V}}_{15}$	0.957	0.961	0.951	0.005	2987
$\bar{\mathcal{V}}_{20}$	0.951	0.963	0.934	0.011	4111
$\bar{\mathcal{V}}_{25}$	0.957	0.975	0.948	0.010	4627
$\bar{\mathcal{V}}_{1,5}$	0.987	0.993	0.980	0.004	1156
$\bar{\mathcal{V}}_{1,10}$	0.976	0.988	0.970	0.006	1917
$\bar{\mathcal{V}}_{1,15}$	0.978	0.989	0.962	0.008	2912
$\bar{\mathcal{V}}_{5,10}$	0.972	0.977	0.956	0.007	2098
$\bar{\mathcal{V}}_{5,15}$	0.971	0.984	0.961	0.008	2813
$\bar{\mathcal{V}}_{10,15}$	0.966	0.977	0.942	0.010	3288
$\bar{\mathcal{V}}_{1,5,10}$	0.980	0.985	0.974	0.004	2132
$\bar{\mathcal{V}}_{5,10,15}$	0.975	0.984	0.952	0.010	2987
$\bar{\mathcal{V}}_{1,5,10,15}$	0.978	0.981	0.971	0.003	2985

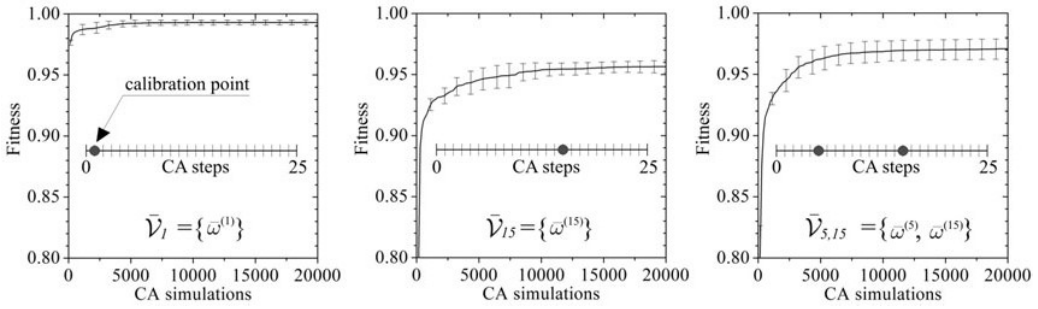

Figure 7. Some convergence plots of the average fitness Θ and its standard deviation for the calibration tests. The statistics were computed over 20 independent runs on each test.

of convergence was very high and the CCPSO algorithm was able to achieve the average fitness $\Theta = 0.993$ with a standard deviation of 0.002 among the different optimization runs (i.e. 0.993 ± 0.002 in the following).

Usually, a lower standard deviation indicates a simpler optimization problem. Because of the probabilistic nature of the CCPSO algorithm, this also implies that a lower number of optimization runs is needed to obtain a reliable calibration.

The other calibrations based on a single map corresponded to more complex optimizations. For example, for $\bar{\mathcal{V}}_{15}$ Fig. 7 shows a decreased speed of convergence (i.e. the initial slope of the plot). Correspondingly, as shown in Table 3, there was a decrease of the average and maximum achieved fitness and a slightly increase in the standard deviation. However, using training sets composed of a single map between $\bar{\omega}^{(15)}$ and $\bar{\omega}^{(25)}$ the complexity of the optimization problem, as detected by the CCPSO algorithm, remained essentially stable. In fact, the Welch's t-tests (Welch 1947) with a significance of 0.05 which we carried out between the achieved fitnesses showed that the results were on average equivalent. It is worth noting that such an equivalence only refers to the achieved fitness considered as a casual variable given by a probabilistic calibration process. As shown later, the used training sets can indeed lead to very different simulation accuracies.

The summary graph in Fig. 8 helps to quantify the effect described above in the case of training sets composed of a single map: up to a certain point, moving forward the calibration point determines a greater difficulty in convergence. In particular, the minimum average fitness of 0.951 was achieved when calibrating using $\bar{\mathcal{V}}_{20}$.

According to Table 3, increasing the size of the training set helped in some cases to achieve a better fitness during the calibration phase. For example, $\bar{\mathcal{V}}_{15}$ led to an average fitness of 0.957 ± 0005 while $\bar{\mathcal{V}}_{5,15}$ led to the fitness of 0.971 ± 0.008. Likely, the positive effect of replacing $\bar{\mathcal{V}}_{15}$ with $\bar{\mathcal{V}}_{5,15}$ is due to the different and more favourable fitness landscape associated with the second training set. However, $\bar{\mathcal{V}}_{5,15}$ and $\bar{\mathcal{V}}_{5,10,15}$ gave statistically equivalent results according to the t-test.

Table 3 also shows the average computation time which was necessary for an optimization. The differences between the elapsed times depend in large part on the different number of CA steps that were necessary for the fitness evaluation. According to Table 3, the average time took by a 25-steps CA simulation was 0.2 s.

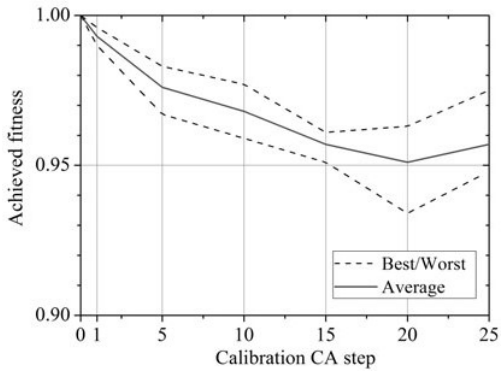

Figure 8. Achieved fitness Θ in the calibration tests based on training sets $\bar{\mathcal{V}}_i$ composed of a single map as a function of the CA step i. The statistics are computed among the 20 optimization runs for each training configuration.

5.2.2. Validations

The main objective of our study was to investigate to what extent the calibration error is affected by the composition of the training set. We measured the accuracy of each calibration as follows: first, using the parameter vector obtained with the calibration, we ran the corresponding CA simulation to calculate the set $\mathcal{V}$ of the configurations at each step (i.e. 25 maps of the area under study); then, we defined the accuracy as the value of function $\Theta\left(\bar{\mathcal{V}}, \mathcal{V}\right)$ described in Section 3, where $\bar{\mathcal{V}}$ is the reference simulation (i.e that obtained using the parameters in Tables 1 and 2).

For each training set, in Table 4 we show the statistics on the accuracy of the corresponding simulations, computed on the 20 independent calibrations. According to the results, the average accuracy varied between 0.909 and 0.974. In this regard, it is important to note that in the literature even small variations of the Kappa index are considered significant in terms of simulation quality (e.g. see Liao *et al.* (2014), García *et al.* (2012), Li *et al.* (2012)).

A first analysis was aimed to establish whether there was a statistically significant difference in the average accuracy obtained with the used training sets. The comparison was conducted trough a rank-based statistical approach, using the R software package (R Core Team 2013) and following the indications given in (Demsar 2006, García *et al.* 2010, Derrac *et al.* 2011). To this purpose, the simulation accuracies obtained using each training set were converted to ranks as follows: first, for each of the 20 initialization seeds, the results obtained through the training sets were ranked from 1 (best result) to 15 (worst result); then, the ranks obtained by each training set were averaged. The mean ranks are shown in Table 4.

Subsequently, we used the Friedman test (Friedman 1937, 1940) to check whether, among the average ranks under comparison, at least two of them represented populations with different median values. Under the null hypothesis, which states equality of all medians between the populations, according to the Friedman statistic we computed the corresponding p-value (i.e. the probabilities of obtaining the sample of average ranks shown in Table 4) which was lower than 0.0001. Thus, assuming a significance level of 0.05, the null hypothesis was rejected.

Then, the average ranks of the considered training sets were compared using the Nemenyi test (Nemenyi 1963, Demsar 2006) with significance of 0.05. For this purpose, the so-called *critical difference CD* was calculated, obtaining the value of 4.80. By definition, two training sets are considered to be statistically equivalent when their average ranks differ by less than the *CD* value.

According to the results in terms of average ranks, depicted in Fig. 9, the simulation accuracy significantly depended on the training set used for calibration.

Among the training sets based on a single map, $\bar{\mathcal{V}}_{10}$ led to the highest average simulation accuracy of 0.963. However, the result provided by $\bar{\mathcal{V}}_{15}$ was equivalent according to the performed statistical test. Instead, the remaining training sets composed of a single map performed significantly worse. For example, the easiest calibration based on $\bar{\mathcal{V}}_1$, in spite of the high value of the achieved average fitness (i.e. $\Theta = 0.993$), led to the low average accuracy of 0.910 (see Table 4). This is not surprising, given that the map produced by a single CA step does not contain enough information on the system's dynamics. Also, the average qualities of the simulations based on $\bar{\mathcal{V}}_5$, $\bar{\mathcal{V}}_{20}$ and $\bar{\mathcal{V}}_{25}$ were all inferior to that of $\bar{\mathcal{V}}_{10}$. It is worth noting that, for $\bar{\mathcal{V}}_{10}$ and $\bar{\mathcal{V}}_{15}$ the average accuracy was quite close to that obtained in the calibration phase. This can also be seen on the graph in Fig. 10, which shows, for the training sets composed of a single map, the average simulation accuracy and achieved fitness as a function of the calibration CA step.

Table 4. Validation statistics in terms of simulation accuracy for the calibration tests. For each training set, the statistics were computed on the 20 set of parameters obtained through the independent calibrations.

Training set	Average	Best	Worst	Std. Dev.	Mean rank
$\bar{\mathcal{V}}_{1}$	0.910	0.936	0.862	0.020	14.6
$\bar{\mathcal{V}}_{5}$	0.929	0.958	0.910	0.015	12.8
$\bar{\mathcal{V}}_{10}$	0.963	0.971	0.956	0.005	6.3
$\bar{\mathcal{V}}_{15}$	0.958	0.960	0.953	0.004	8.3
$\bar{\mathcal{V}}_{20}$	0.935	0.957	0.910	0.015	11.7
$\bar{\mathcal{V}}_{25}$	0.928	0.957	0.907	0.015	13.4
$\bar{\mathcal{V}}_{1,5}$	0.946	0.964	0.918	0.016	10.7
$\bar{\mathcal{V}}_{1,10}$	0.950	0.963	0.934	0.008	10.3
$\bar{\mathcal{V}}_{1,15}$	0.962	0.975	0.936	0.010	5.8
$\bar{\mathcal{V}}_{5,10}$	0.962	0.970	0.954	0.005	6.8
$\bar{\mathcal{V}}_{5,15}$	0.975	0.983	0.968	0.005	1.9
$\bar{\mathcal{V}}_{10,15}$	0.965	0.972	0.946	0.007	5.3
$\bar{\mathcal{V}}_{1,5,10}$	0.959	0.971	0.946	0.008	6.9
$\bar{\mathcal{V}}_{5,10,15}$	0.974	0.983	0.967	0.005	2.4
$\bar{\mathcal{V}}_{1,5,10,15}$	0.970	0.974	0.963	0.003	3.0

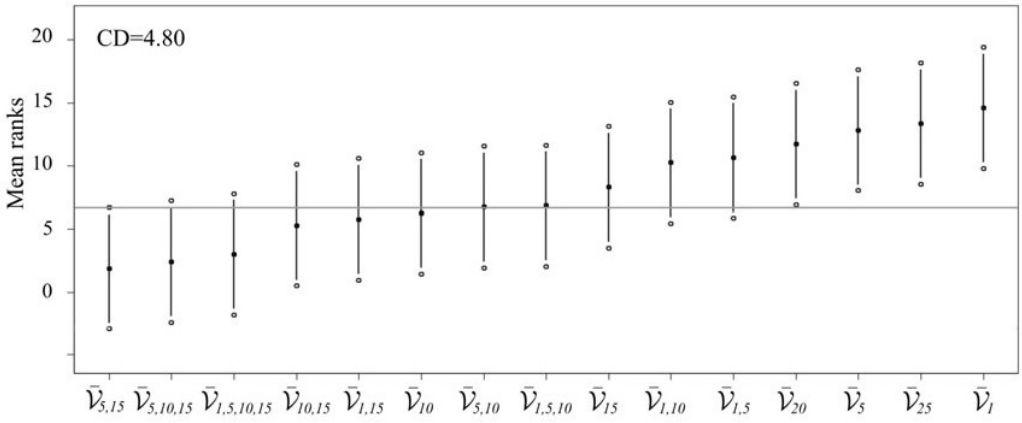

Figure 9. Comparison between the mean ranks achieved by the considered training sets. A lower rank corresponds to a higher simulation accuracy. The vertical bars represent the critical difference $CD = 4.80$ calculated using the Nemenyi test with significance of 0.05. The filled circles below the horizontal line represent training sets leading to simulations that on average are equivalent to those of best training set (i.e. $\bar{\mathcal{V}}_{5,15}$).

According to Table 4, among the training sets composed of two maps, $\bar{\mathcal{V}}_{5,15}$ provided simulations with the highest average accuracy of 0.975, which is also the best result among all the considered training sets. Compared to $\bar{\mathcal{V}}_{5,15}$, the training sets $\bar{\mathcal{V}}_{10,15}$ and $\bar{\mathcal{V}}_{1,15}$ were equivalent according to the Nemenyi test.

Also, as shown in Fig. 9, using $\bar{\mathcal{V}}_{5,10,15}$ produced simulations that were equivalent to those obtained using the smaller training set $\bar{\mathcal{V}}_{5,15}$. Therefore, increasing the latter with an intermediate map proved essentially useless. Even the largest training set $\bar{\mathcal{V}}_{1,5,10,15}$ did not help to improve the quality of the optimized CA model.

Interestingly, the comparisons also showed that incrementing through an additional map $\bar{\mathcal{V}}_{10}$, which is the best training set composed of a single map, did not result in sig-

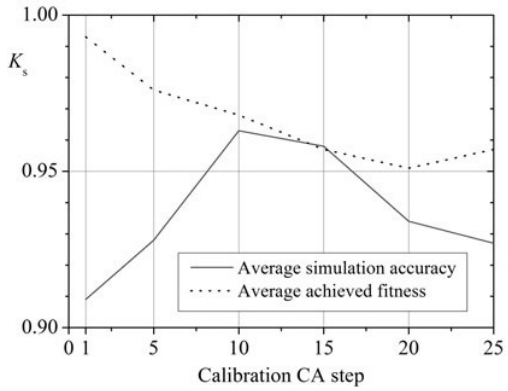

Figure 10. Average simulation accuracy and achieved fitness for the different calibrations based on a single map (the statistics were computed on the 20 independent calibrations).

nificant improvements in the simulations. In fact, $\bar{\mathcal{V}}_{1,10}$, $\bar{\mathcal{V}}_{5,10}$, and $\bar{\mathcal{V}}_{10,15}$ were equivalent to $\bar{\mathcal{V}}_{10}$.

5.2.3. Analysis of accuracy during the simulations

The above comparisons are based on a measure of accuracy concerning the entire CA simulation of 25 steps. However, it may also be interesting to examine how the accuracy varies during the simulation. To this end, a further analysis of the calibration outcomes was conducted using the best-parameter vectors obtained with the different training sets. This makes sense because the typical calibration procedure based on a probabilistic search consists of running several independent optimization processes to choose the best result in terms of fitness. In particular, for each best-parameter vector we ran the CA simulations to calculate, at each step, the measure of agreement K_s between the simulated and the reference configurations.

The values of K_s along the simulation for the different training sets are depicted in Fig. 11. According to the latter, the calibration based on $\bar{\mathcal{V}}_1$ exhibits a rapid fall in accuracy immediately after the calibration point. The same behaviour, although less pronounced, was found using $\bar{\mathcal{V}}_5$. Instead, for $\bar{\mathcal{V}}_{10}$ and $\bar{\mathcal{V}}_{15}$ the evolution of accuracy during the simulation showed a low variability and quite high values. Moving farther the calibration point led to a worsening of the accuracy curve. In particular, $\bar{\mathcal{V}}_{25}$ produced a good agreement only in the immediate vicinity of the initial and final configurations. By analysing Fig. 11 we recognize that even a single calibration point can produce simulations of reasonable quality, provided it is not too close to the beginning or the end of simulation.

A similar behaviour can be observed in the cases of larger training sets. For example, using $\bar{\mathcal{V}}_{1,5}$ and $\bar{\mathcal{V}}_{1,10}$ a relatively high agreement with the validation set was limited to about the first 10 of the 25 steps of simulation. Slightly better results were obtained using $\bar{\mathcal{V}}_{1,15}$.

Looking at the cases of training sets composed of two maps, it appears that when different calibration points are well spaced and not too close to the beginning of the simulation, the accuracy is quite high and steady until the final step. In fact, the best results were obtained for $\bar{\mathcal{V}}_{5,15}$ and, to a lesser extent, for $\bar{\mathcal{V}}_{5,10}$ and $\bar{\mathcal{V}}_{10,15}$.

5.2.4. A forecasting exercise

As a final analysis, we examined the above results from a land-use-change forecasting perspective: assuming that the current time corresponds to the 15th CA step and that

we had to predict the configuration $\bar{\omega}^{(25)}$ at the 25th CA step, which of the above training sets would be better? Given the assumption on the constraints, the considered test problem corresponds to a prediction of the spatial scenario after 20 years. Obviously, we assumed that the current configuration is available for the training set.

The results, expressed in terms of K_s at the 25th CA step, are shown in Fig. 12. Also, in Table 5 for the best and the worst training sets, we show: value of K_s, *number of hits* (land-use changes in the reference configuration $\bar{\omega}^{(25)}$ which were correctly predicted), *number of misses* (land-use changes in $\bar{\omega}^{(25)}$ incorrectly predicted as persistence or different change), *false alarms* (land-use persistence in $\bar{\omega}^{(25)}$ incorrectly predicted as

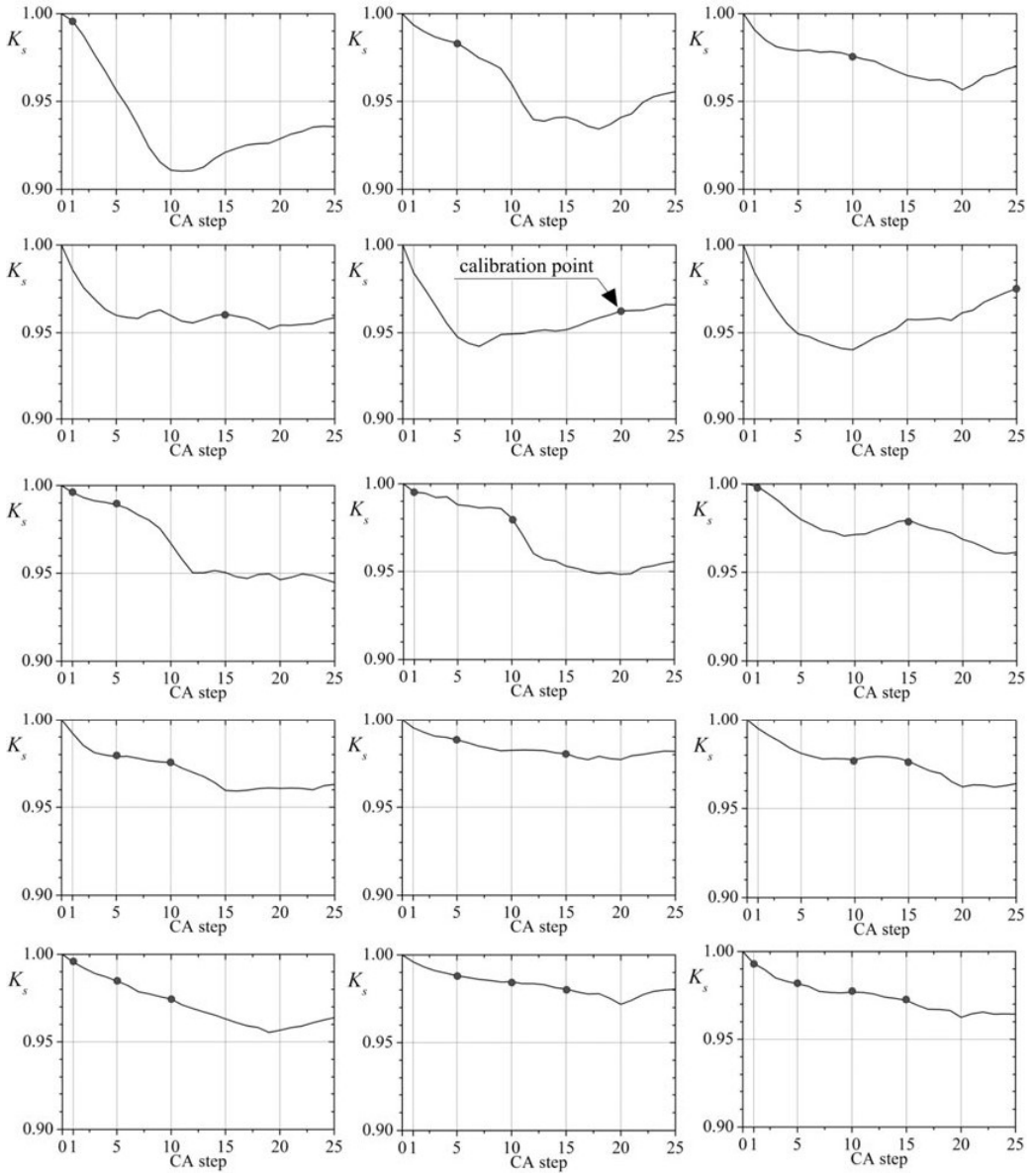

Figure 11. Value of K_s during the simulation for the different calibrations. For each training set, we used the vector of parameters corresponding to the best achieved fitness.

change) and *percentage of hits* (computed as hits over the total change in the reference configuration).

As can be seen, the best forecasting ability was achieved through a model calibrated using two configurations (i.e. $\bar{\omega}^{(5)}$ and $\bar{\omega}^{(15)}$). However, the calibration with $\bar{\mathcal{V}}_{5,10,15}$ was essentially equivalent. The poor performance of the training set $\bar{\mathcal{V}}_{1,5,10,15}$ is due to the inclusion of configuration $\bar{\omega}^{(1)}$, which proves to worsen the predictive abilities of the model. In this case, we can recognize a kind of overfitting that creates a bias in the calibration towards the initial part of the simulation.

For the best (i.e. $\bar{\mathcal{V}}_{5,15}$) and the worst (i.e. $\bar{\mathcal{V}}_{15}$) forecasting ability of the model, in

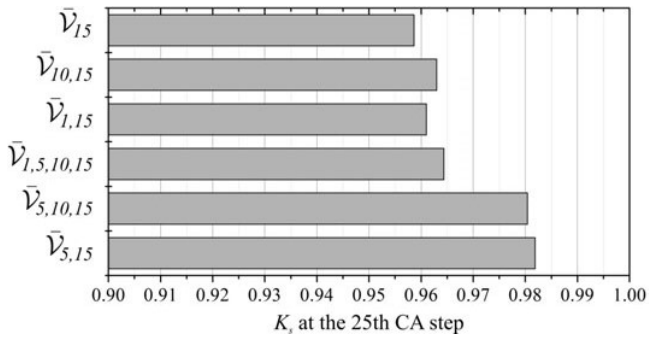

Figure 12. Comparison between different calibration options tested in the forecasting exercise. For each training set we used the vector of parameters corresponding to the best achieved fitness during the calibration process.

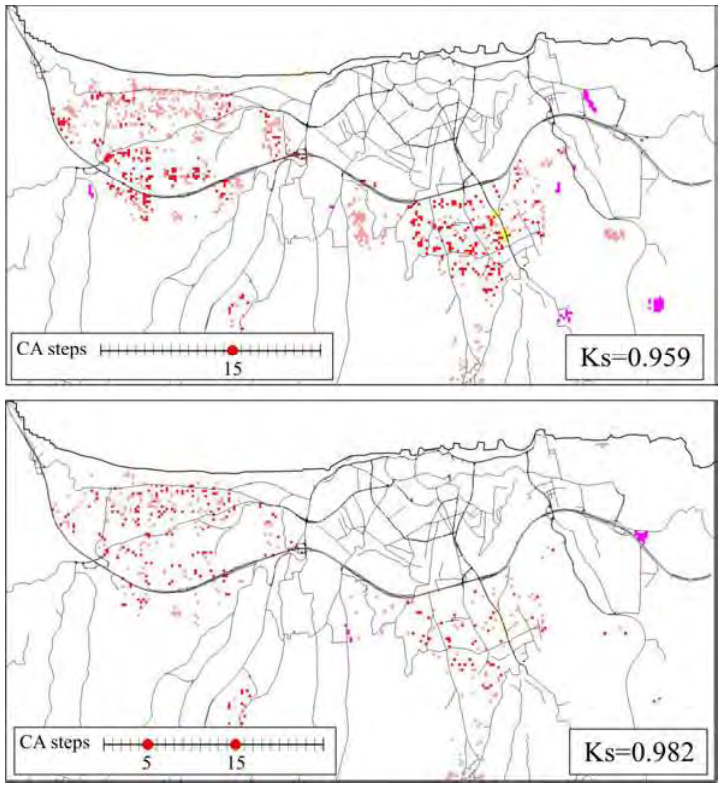

Figure 13. Forecasting exercise: maps of the differences between the final simulated configuration (i.e. $\omega^{(25)}$) and the final reference configuration (i.e. $\bar{\omega}^{(25)}$). The calibration points are highlighted.

Figure 13 we show the maps of the differences between the final simulated configuration (i.e. $\omega^{(25)}$) and the final reference configuration (i.e. $\bar{\omega}^{(25)}$). This with the aim of helping to visualize the effect of calibration errors in the case under study.

According to Table 5, adding the map $\bar{\omega}^{(5)}$ to the training set $\bar{\mathcal{V}}_{15}$ increased by 6.1% the percentage of hits.

6. Conclusions and Future Work

The automatic calibration of urban CA models is a crucial phase in ensuring the reliability and accuracy of simulations. However, calibration can be a challenging problem, since it requires availability of data, efficient and effective optimization algorithms and significant computational resources.

In this article, using a suitable experimental setup and some advanced computational techniques, we investigated the influence of the composition of the training set on the quality of the resulting calibration. In particular, in spite of the substantial equivalence of most of the fitness values obtained during the calibrations, a suitable statistical test revealed several significant differences in the quality of the corresponding simulations.

A first group of indications that can be drawn from the above results concerns the use of a single historical map for calibration. In this respect, the experiments showed that:

- training sets including only calibration points very close to the beginning of the simulation (i.e. $\bar{\mathcal{V}}_1$ and $\bar{\mathcal{V}}_5$) led to easy optimization problems (i.e. high achieved fitness). However, the corresponding accuracy of simulations was poor.
- training sets composed of a single calibration point quite far from the beginning of the simulation (i.e. $\bar{\mathcal{V}}_{20}$ and $\bar{\mathcal{V}}_{25}$) led to more difficult optimization problems and again to poor accuracy of simulations;
- in contrast, a reasonable quality of the calibrated model was obtained using a single historical map referring to a point which is not too close nor too far from the beginning of the simulation (i.e. $\bar{\mathcal{V}}_{10}$ or $\bar{\mathcal{V}}_{15}$).

Other insights that can be derived from the results discussed above concern the use of multiple maps in the calibration phase. In this regard, the computational experiments showed that:

- in most cases, incrementing the training set through additional maps did not produce significant improvements in the simulation accuracy; this was observed when the added map was not able to bring additional informational content (e.g. $\bar{\omega}_5$ added to $\bar{\mathcal{V}}_{10}$ or $\bar{\omega}_{10}$ added to $\bar{\mathcal{V}}_{5,15}$);
- increasing the size of the training set to more than two maps has never led to significant improvements in simulation accuracy;
- the best training set was composed of two calibration points (i.e. at the CA steps 5 and 15), which were well spaced and not too close to the beginning of the simulation.

It is important to highlight that the insights that can be drawn from the experiments described above refer to a case in which the rules that determine the urban dynamics (i.e. the parameters) are constant over time. This can be, at least approximately, the

Table 5. Forecasting exercise: accuracy in simulating the final configuration through the best and the worst training sets.

Training set	K_s	Hit [cells]	Missed [cells]	False positive [cells]	Hit [%]
$\bar{\mathcal{V}}_{15}$	0.959	9610	1157	63	89.3
$\bar{\mathcal{V}}_{5,15}$	0.982	10274	493	79	95.4

case for many historical periods and urban areas in the world. Nevertheless, if the mutual influences between the different land uses or those exerted by the driving factors slightly vary over time, we could still apply the CA described in section 2, as well as the calibration procedure, accepting a certain model error. In this case, in contrast to the experiments discussed above, more likely an additional map in the training set would bring a significant contribution of information on the dynamics of the system, favouring the convergence towards a better vector of parameters.

Our computational study has some more limitations, which we plan to overcome in the future. A first step would be to repeat the investigation for a significant number of random urban dynamics, rather than just for one. Obviously, this would increase considerably the number of required CA simulations.

There are also other aspects that deserve to be investigated. For example, to what extent a different fitness function could positively influence the quality of calibration. In fact, there are many alternatives to the agreement measure between maps that we used, including multi-objective fitness functions based on various landscape metrics (Li *et al.* 2013). In conclusion, on the issue of calibrating CAs for the simulation of urban dynamics, there is still room for useful research.

Acknowledgement

We gratefully acknowledge the support of nVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

Avolio, M.V., *et al.*, 2006. SCIARA γ2: An improved cellular automata model for lava flows and applications to the 2002 Etnean crisis. *Computers & Geosciences*, 32 (7), 876–889.

Barredo, J.I., *et al.*, 2003. Modelling dynamic spatial processes: simulation of urban future scenarios through cellular automata. *Landscape and Urban Planning*, 64 (3), 145 – 160.

Batty, M. and Xie, Y., 1994. From cells to cities. *Environment and Planning B*, 21, 31–48.

Blecic, I., *et al.*, 2014a. Urban metabolism and climate change: A planning support system. *Int. J. Applied Earth Observation and Geoinformation*, 26, 447–457.

Blecic, I., *et al.*, 2004. Modelling Urban Dynamics with Cellular Automata: a Model of the City of Heraklion. *In*: *7th AGILE Conference on Geographic Information Science* University of Crete Press, 313–323.

Blecic, I., Cecchini, A., and Trunfio, G.A., 2010. A Comparison of Evolutionary Algorithms for Automatic Calibration of Constrained Cellular Automata. *In*: *ICCSA 2010, Part I*, Vol. 6016 of *LNCS* Springer, 166–181.

Blecic, I., Cecchini, A., and Trunfio, G.A., 2013. Cellular automata simulation of urban dynamics through GPGPU. *The Journal of Supercomputing*, 65 (2), 614–629.

Blecic, I., Cecchini, A., and Trunfio, G.A., 2014b. Fast and Accurate Optimization of a GPU-accelerated CA Urban Model through Cooperative Coevolutionary Particle Swarms. *Procedia Computer Science*, 29, 1631 – 1643.

Cecchini, A., 1996. Urban modelling by means of cellular automata: generalised urban

automata with the help on-line (AUGH) model. *Environment and Planning B*, 23, 721–732.

Cheng, J. and Masser, I., 2004. Understanding spatial and temporal processes of urban growth: cellular automata modelling. *Environment and Planning B: Planning and Design*, 31 (2), 167–194.

Clarke, K.C., Hoppen, S., and Gaydos, L., 1997. A self-modifying cellular automaton model of historical urbanization in the San Francisco Bay area. *Environment and Planning B: Planning and Design*, 24 (2), 247–261.

Cohen, J., 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20 (1), 37–46.

Demsar, J., 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1–30.

Derrac, J., *et al.*, 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1 (1), 3–18.

Engelen, G. and White, R., 2008. Validating and Calibrating Integrated Cellular Automata Based Models of Land Use Change. *In*: S. Albeverio, D. Andrey, P. Giordano and A. Vancheri, eds. *The Dynamics of Complex Urban Systems*. Physica-Verlag HD, 185–211.

Engelen, G., White, R., and de Nijs, T., 2005. Environment Explorer: Spatial Support System for the Integrated Assessment of Socio-Economic and Environmental Policies in the Netherlands. *Integrated Assessment*, 4 (2).

Feng, Y., *et al.*, 2011. Modeling dynamic urban growth using cellular automata and particle swarm optimization rules. *Landscape and Urban Planning*, 102 (3), 188 – 196.

Friedman, M., 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32 (200), 675–701.

Friedman, M., 1940. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *The Annals of Mathematical Statistics*, 11 (1), 86–92.

García, A.M., *et al.*, 2012. A comparative analysis of cellular automata models for simulation of small urban areas in Galicia, NW Spain. *Computers, Environment and Urban Systems*, 36 (4), 291–301.

García, S., *et al.*, 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.*, 180 (10), 2044–2064.

Geertman, S. and Stillwell, J., 2004. Planning support systems: an inventory of current practice. *Computers, Environment and Urban Systems*, 28 (4), 291 – 310.

Goldstein, N.C., 2003. Brains vs. Brawn-Comparative strategies for the calibration of a cellular automata-based urban growth model. *In*: *Proceedings of the 7-th International Conference on GeoComputation, 8-10 September*.

He, C., *et al.*, 2008. Modelling dynamic urban expansion processes incorporating a potential model with cellular automata. *Landscape and Urban Planning*, 86 (1), 79 – 91.

Kennedy, J. and Clerc, M., 2006. Standard PSO 2006. [online] Available from: http://www.particleswarm.info/Standard_PSO_2006.c [Accessed Jan 2014].

Kennedy, J. and Eberhart, R., 1995. Particle swarm optimization. *In*: *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Vol. 4, 1942–1948 vol.4.

Lavalle, C., *et al.*, 2011. A high resolution land use/cover modelling framework for

Europe: introducing the EU-ClueScanner100 model. *In*: *ICCSA 2011*, LNCS, Santander, Spain Berlin, Heidelberg: Springer-Verlag, 60–75.

Li, X., *et al.*, 2013. Calibrating cellular automata based on landscape metrics by using genetic algorithms. *International Journal of Geographical Information Science*, 27 (3), 594–613.

Li, X. and Yeh, A.G.O., 2000. Modelling sustainable urban development by the integration of constrained cellular automata and GIS. *International Journal of Geographical Information Science*, 14 (2), 131–152.

Li, X., *et al.*, 2012. Assimilating Process Context Information of Cellular Automata into Change Detection for Monitoring Land Use Changes. *Int. J. Geogr. Inf. Sci.*, 26 (9), 1667–1687.

Liao, J., *et al.*, 2014. A neighbor decay cellular automata approach for simulating urban expansion based on particle swarm intelligence. *International Journal of Geographical Information Science*, 28 (4), 720–738.

Nemenyi, P., 1963. Distribution-free multiple comparisons. Thesis (PhD). New Jersey, USA.

Omidvar, M.N., *et al.*, 2010. Cooperative Co-evolution for large scale optimization through more frequent random grouping.. *In*: *Proceedings of the IEEE Congress on Evolutionary Computatio* IEEE, 1–8.

Potter, M.A. and De Jong, K.A., 1994. A Cooperative Coevolutionary Approach to Function Optimization. *In*: *Parallel Problem Solving from Nature*, Vol. 866 of *Lecture Notes in Computer Science* Springer-Verlag, 249–257.

R Core Team, 2013. R: A Language and Environment for Statistical Computing. [online] Available from: http://www.R-project.org/ [Accessed Sept 2014].

Rabbani, A., Aghababaee, H., and Rajabi, M.A., 2012. Modeling dynamic urban growth using hybrid cellular automata and particle swarm optimization. *Journal of Applied Remote Sensing*, 6 (1).

Santé, I., *et al.*, 2010. Cellular automata models for the simulation of real-world urban processes: a review and analysis. *Landscape and Urban Planning*, 96 (2), 108–122.

Straatman, B., White, R., and Engelen, G., 2004. Towards an automatic calibration procedure for constrained cellular automata. *Computers, Environment and Urban Systems*, 28 (1-2), 149–170.

Trunfio, G.A., 2014. Enhancing the firefly algorithm through a cooperative coevolutionary approach: an empirical study on benchmark optimisation problems. *IJBIC*, 6 (2), 108–125.

van Vliet, J., Bregt, A.K., and Hagen-Zanker, A., 2011. Revisiting Kappa to account for change in the accuracy assessment of land-use change models. *Ecological Modelling*, 222 (8), 1367 – 1375.

vandenBergh, F. and Engelbrecht, A.P., 2004. A Cooperative Approach to Particle Swarm Optimization. *IEEE Trans. Evolutionary Computation*, 8 (3), 225–239.

Welch, B.L., 1947. The Generalization of 'Student's' Problem when Several Different Population Variances are Involved. *Biometrika*, 34 (1/2), 28–35.

White, R. and Engelen, G., 1993. Cellular automata and fractal urban form: A cellular modeling approach to the evolution of urban land-use patterns. *Environment and Planning A*, 25, 1175–1199.

White, R. and Engelen, G., 2000. High-resolution integrated modelling of the spatial dynamics of urban and regional systems. *Computer, Environment and Urban Systems*, 24, 383–400.

White, R., Engelen, G., and Uljee, I., 1997. The use of constrained cellular automata for

high-resolution modelling of urban land-use dynamics. *Environment and Planning B: Planning and Design*, 24 (3), 323–343.

Wu, F., 1998. SimLand: A Prototype to Simulate Land Conversion Through the Integrated GIS and CA with AHP-Derived Transition Rules. *International Journal of Geographical Information Science*, 12 (1), 63–82.

Yang, Z., Tang, K., and Yao, X., 2008. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178 (15), 2985–2999.

Yeh, A.G.O. and Li, X., 2006. Errors and uncertainties in urban cellular automata. *Computers, Environment and Urban Systems*, 30 (1), 10 – 28.