

Parameter Calibration of the Patch Growing Algorithm for Urban Land Change Simulations

Rodrigo Lopez-Farias

CONACyT - Centro de Investigación
en Ciencias de Información
Geoespacial A.C.
Queretaro, Mexico
rlopez@centrogeo.edu.mx

S. Ivvan Valdez

CONACyT - Centro de Investigación
en Ciencias de Información
Geoespacial A.C.
Queretaro, Mexico
svaldez@centrogeo.edu.mx

A. Garcia-Robledo

CONACyT - Centro de Investigación
en Ciencias de Información
Geoespacial A.C.
Queretaro, Mexico
agarcia@centrogeo.edu.mx

Abstract—Urban growth modelling is a current trend in geo-computation due to its impact on the local living environment and the quality of life. The FUTURE Urban-Regional Environment Simulation (FUTURES) model produces projections of landscape patterns by coupling land suitability, per-capita demand, and patch growing algorithm (PGA) sub-models. In particular, the PGA is the urban growing simulator component that takes into account the stochastic nature of urban development. It requires a set of parameters, namely *compactness mean*, *compactness range*, and *discount factor* to approximate the general characteristics of the urban development structure. The fitness of the parameters is measured by computing the difference between the area and compactness histograms of the observed and simulated urban growths. On the one hand, the authors find these parameters via an exhaustive grid search; nevertheless, this requires evaluating all the points in the grid, which implies a high computational cost because each point is associated with several PGA simulations. In addition, the approximation is limited to be one of the points in the grid. Thus, the better the precision is, the higher the required computational cost. On the other hand, evolutionary algorithms have been widely used for the automatic calibration of parameters but, in general, they are not designed to use a low number of evaluations, require expert tuning, and to define the stop criteria. Therefore, we propose an algorithm to find the adequate parameters, with minimum expert intervention, using an Estimation of Distribution Algorithm designed to use a low number of function evaluations (EDALNFE) when compared to the grid search. EDALNFE provides several assets: it delivers competitive results, it requires a low number of function evaluations, it does not require expert settings, and it is equipped with an automatic stop criterion. The proposed algorithm is compared to exhaustive grid search (the only method readily available in the FUTURES package) and differential evolution to demonstrate its superior performance.

Index Terms—Land Use Change, Estimation of Distribution Algorithm, Patch Growing Algorithm, Optimization

I. INTRODUCTION

FUTURE Urban-Regional Environment Simulation (FUTURES) is a framework designed to produce consistent urban growth projections given a set of initial geographic conditions. It is one of the most accessible and used urban growth simulation models and it has been studied and compared against other popular models [1].

FUTURES is a modular model composed by three major modules: 1) the Land Use Suitability function, which produces

the initial probability of change from non-urban to urban land use; 2) the land demand per-capita, which estimates the quantity of land that should be allocated in the space; and 3) the Patch Growth Algorithm (PGA), which incorporates the stochastic nature of urban growth to simulate the allocation of the estimated urban land units.

Although the framework has the potential to give significant insights about the urbanization structure and spatial trends, the PGA parameters are not straight-forward to obtain for generating future urban scenarios. The available framework allows to select these parameters by expert experimentation which is, basically, a trial-error strategy.

Therefore we propose a competitive novel population-based meta-heuristic EDALNFE to automatically approximate global optimum parameters in an efficient way.

II. RELATED WORK

The calibration of an urban growth model seeks to determine the parameter values that most accurately represent the observed land use [1] by searching for a combination of parameter values that minimizes the error between the simulated patch changes and the observed patch changes.

There are as many calibration algorithms as there are urban growth models since each model has different parameters to calibrate. Nonetheless, we can identify two rough classes of calibration strategies: brute-force-based calibration and optimization-based calibration. These two classes are discussed below.

A. Brute-force Calibration

Brute-force calibration consists of testing all the possible combinations of sets of parameter values. In a worst-case scenario this is equivalent to a factorial experiment design.

In [2] the authors calibrate the widely-used SLEUTH urban growth model by following a brute-force approach. They test a set of combinations of the control parameters. Results are then sorted, and the parameters with the best results are used to begin the next calibration round.

In the case of FUTURES the plugin for GRASS includes the `r.futures.calib` module, which runs the PGA model (detailed in the next section) with different combinations

of parameters: compactness mean, compactness range, and discount factor. The structure of the simulated urban patches (land sub-regions composed by contiguous cells) of these runs are then compared against the observed patch changes in the ground truth to select the best set of parameters in terms of the measured error [3].

The brute-force approach is clearly highly computational intensive for a thorough parameter calibration. This approach usually requires the use of parallel computers. Also, often the search space of the continuous parameters is discretized. Hence, no values out of the discrete set are ever tested.

B. Optimization-based Calibration

Optimization-based calibration is performed by meta-heuristic algorithms which explore the space of parameter values by optimizing an objective function. A common class of meta-heuristics used for urban growth model calibration is population meta-heuristics such as Genetic Algorithms (GAs).

The disadvantage of optimization-based calibration is the need of adequately setting the meta-heuristic parameters, which can represent a non-trivial task. For example, in the case of GAs this involves designing an objective function, choosing a chromosome representation, generating an initial population, as well as defining the selection, crossover, and mutation operators.

The advantages include the reduction of number of parameters tests which often implies significantly lower calibration times when compared to the brute-force approach. Also, unlike brute-force algorithms, meta-heuristics can efficiently search parameter values in the continuous space. Following, we review the most iconic research in meta-heuristic algorithms for urban growth model calibration.

In [4] it is presented a memetic algorithm to calibrate a Dinamica-EGO-based LULC urban patch growth model. The memetic algorithm is used to heuristically search the parameter space by “extending a genetic algorithm with a local search function”.

In [5] is presented a genetic algorithm to calibrate the transition rules of a cellular automata urban growth model based on the analysis of multi-temporal satellite imagery and population density.

In [6] a genetic algorithm is proposed to calibrate the SLEUTH growth algorithm on data for the Azadshahr, Gonbadekavoos, and Gorgan cities in Iran. The authors conclude that the GA prevents premature convergence to a local minimum, and favors the approximation to a global minimum.

In [7] it is proposed a cellular automata-based urban growth model that uses an Artificial Bee Colony optimization algorithm (ABC) to extract transition rules. The calibration is conducted through an Ant Colony Optimization algorithm (ACO) and compared against similar swarm-intelligence methods. Experiments were conducted on urban growth data from Urmia, Iran. Authors conclude that the ABC algorithms outperforms the ACO-based model “with fewer quantity and allocation errors”.

In [8] authors optimize the parameters of a patch-based cellular automata model by using a GA that optimizes the size, shape, and compactness of the urban growth model. A surrogate model is used in the GA to reduce the number of the objective function evaluations.

In [9] authors propose a cellular automata model that uses irregular cells for simulating land-use change. They calibrate their model with a particle swarm algorithm that optimizes the agreement between the reference and simulation maps measured in their contingency matrix.

In [10] authors try two optimization algorithms for the calibration of the Fuzzy Cellular Urban Growth Model (FCUGM): a GA and parallel annealing. They conclude that the GA produces a better calibrated model.

In [11] authors calibrate an urban cellular automata model in two steps. First, they reduce the number of parameters to calibrate by determining the weights of the suitability evaluation factors with logistic regression. Then, they use a GA to calibrate the remaining parameters.

To the best of our knowledge, our work presents the first optimization-based calibration approach that specifically targets the FUTURES patch-growing algorithm.

III. A BRIEF OVERVIEW OF FUTURES FRAMEWORK

FUTURES is a framework that simulates urban land change and urban growth using three modules: the potential module for computing land-use-change suitability, the one for the estimation of land demand, and the patch growing algorithm (PGA).

a) *Land-use-change Suitability*: It measures the likelihood of new land development based on the association of the socioeconomic, infrastructural, and environmental dimensions of a region. It is modeled with a logistic regression to associate geographic variables with urban change. It is described with Equation (1):

$$s'_i = a + \sum_{h=0}^m \beta_h \cdot x_{ih} \quad (1)$$

where s'_i is the preliminary suitability score computed as the result of a linear combination of the m map predictors coefficients β_i indexed by h that multiply each map variable x_h at cell i plus the intercept a .

The preliminary suitability score s'_i is mapped with the sigmoid function $s_i = e^{s'_i} / (1 + e^{s'_i})$ to produce the suitability map. The suitability map is the most important map in the PGA to simulate realistic urban scenarios of the spatial urban growth and deserves an independent study.

b) *Demand*: It is a module that predicts the land demand per-capita given a population-size.

c) *PGA*: It is the module of main interest to this work. It simulates the urban growth dynamics given a suitability map of urban land change (Equation (1)), parameters Θ , and land demand estimation, considering a stochastic process of urban land change. Every developed cell by PGA modifies

the suitability score in the neighborhood s'_i using Equation (2)

$$s_i = s'_i * d^{-\alpha}, \quad (2)$$

where α is a number from a uniform probability distribution that controls the attraction of the urban development around a given cell. The uniform probability distribution is defined by the compactness mean μ_c , compactness range μ_σ parameters and the uniform random variable $r \in [0, 1)$:

$$\alpha = (\mu_c - \mu_\sigma) \cdot 0.5 + r \cdot \mu_\sigma. \quad (3)$$

The basic steps of PGA are:

- 1) Select randomly an undeveloped cell in the suitability map.
- 2) Set the selected cell as developed with a probability given by the suitability map.
- 3) If the cell is not set as developed, then repeat Step 1.
- 4) PGA explores the new urban land suitability of the neighbors. The cell is added to the growing patch depending on the composite suitability score s_i .
- 5) The composite scores of the neighboring cells are computed and sorted.
- 6) The PGA ranks the candidate cells according to their composite score s_i to guide the patch growth through the neighborhood search. It stops when it meets a stop criterion, such as a maximum patch size.

The PGA requires the initial suitability map S , the initial urban map M_t , and the estimated land demand for the next time step $\Delta\hat{Q}_{t+1}$.

A. Optimization Problem Definition

To define the optimization problem, it is important to consider the statistical representation of the urban land structure.

The land structure is represented by small land sub-regions composed by contiguous cells called patches. The set of all patches found in an urban map are characterized according to density estimations of patch shapes and sizes.

The density estimation of patch sizes H_{sz} is calculated from the histogram of patch sizes. The patch size is the number of contiguous developed cells considering the 4-neighbor rule [12].

Similarly, the density estimation of patch shapes H_{sh} is calculated from the histogram of shapes of the patches sh . The shape is the quotient of the perimeter of the observed patch j of size z and the perimeter of a circle of the same size, as described in Equation (4).

$$pshape = \frac{\text{perimeter of patch } j \text{ with size } z}{\text{perimeter of circle of size } z}. \quad (4)$$

Where if $sh > 1$ means that the observed patch is not as compact as the circle.

The problem is to find the best PGA parameters, namely compactness mean μ_c , compactness range σ_c , and discount factor φ represented by $\Theta = \{\mu_c, \sigma_c, \varphi\}$ that produce a likely simulated increment of urban map $\Delta\hat{M}_{t+1}$. The parameters μ_c, μ_σ are described in Equation (2), and φ is a discount factor

that shrinks the initial patch sizes before calculating its density distribution, in order to compensate the smaller size of the new developed patches.

We want to produce the map $\Delta\hat{M}_{t+1}$ presenting similar size and patch shape density estimations to the urban map M_t , given an estimated land demand \hat{Q}_{t+1} , and initial urban and suitability maps M_t and S_t , respectively. Equation (5), describes the stochastic dynamic process for simulating land-use change.

$$\hat{M}_{t+1} = f_{PGA}(M_t, \{S_t, \hat{Q}_{t+1}, \Theta\}), \quad (5)$$

where \hat{M}_{t+1} is the estimation of the urban growth map at time $t + 1$. f_{PGA} is the dynamic process that receives the initial urban map M_t , suitability map S_t , and parameters $\Theta = \{\mu_c, \mu_\sigma\}$.

The urban land increment $\Delta\hat{M}_{t+1}$ is a map difference used in the objective function defined by:

$$\Delta\hat{M}_{t+1} = \hat{M}_{t+1} - M_t. \quad (6)$$

In this regard, the minimization problem is given by:

$$\arg \min_{\Theta} d_{sz}(\Theta) + d_{sh}(\Theta), \quad (7)$$

where $d_{sz}(\Theta)$ and $d_{sh}(\Theta)$ are the density distribution distances for the size and shape defined by:

$$\begin{aligned} d_{sz}(\Theta) &= d(H_{sz}(psize_t), H_{sz}(\varphi \cdot psize_{t+1})) \\ d_{sh}(\Theta) &= d(H_{sh}(pshape_t), H_{sh}(pshape_{t+1})) \end{aligned} \quad (8)$$

where φ is the discount factor, $psize_t$, $psize_{t+1}$ and $pshape_t$, $pshape_{t+1}$ are the lists of simulated and observed patch sizes and shapes in association with maps M_t and $\Delta\hat{M}_{t+1}$. The density distribution distance is defined as:

$$d(\mathbf{a}, \mathbf{b}) = 0.5 \cdot \sum_{i=1}^{bins} \frac{(a_i - b_i)^2}{a_i + b_i}, \quad (9)$$

where a_i and b_i are the bin values of two density distributions H multiplied by 100. The bins with no values are discarded to compute the distance.

IV. OPTIMIZATION METHODS

Evolutionary algorithms have been widely used for derivative-free and black-box optimization [13]. Parameter calibration of the PGA model is addressed with this kind of algorithms because of the complexity of the simulation algorithms that are, in this case, black boxes, and the characteristics of the objective functions that are distances in non-derivable measures. In this regard Differential Evolution (DE) [14] has been demonstrated competitive results for the calibration of water and rainfall models among others in recent research [15]. In the same regard, estimation of Distribution Algorithms (EDAs) have been used for model calibration and optimization [16]. Nevertheless, they have not been applied to the patch growing problem or to geography-related problems. EDAs use

a explicit probability distribution that could be equipped with mechanisms for improving the search or the convergence. In this case, we introduce a new EDA with two main advantages: it requires a low number of objective function evaluations and it presents fast convergence in contrast with the grid search and the DE. In contrast to other evolutionary algorithms, the EDA convergence can be measured and used as a stop criterion.

Additionally, the proposed EDA, named Estimation of Distribution Algorithm for a Low Number of Function Evaluations (EDALNFE), has a unique input parameter, hence it does not requires of expert settings.

In this section, we present: 1) a description of the grid search, that basically is an exhaustive search over a discretized search space; 2) the Differential Evolution, an evolutionary algorithm that reports competitive results for model calibration; and 3) the proposed EDALNFE, an Estimation of Distribution Algorithm specifically designed for this calibration task.

A. Grid Search

The grid search works as follows. Each variable domain is partitioned into $N_{\mu c}$, $N_{\sigma c}$, and $N_{\mu area}$ discrete points that are equally spaced nodes of the grid. Then, every node is a combination of parameters, so that the combination reporting the minimum objective function value is the best. On the one hand, this strategy presents the advantage of searching in representative solutions of the whole search space. On the other hand, it only looks at predefined configurations. Hence, very likely, the optimal parameters do not belong to the grid. Nevertheless, the grid search serves as the baseline result in computation cost and as objective function value. The aim of this research is to present a proposal that delivers a configuration with a lower objective function value than the grid search with a similar number of objective function evaluations, or a similar objective function value with a lower number of function evaluations.

The specific discrete points of the grid within the search limits are defined in Equation (11).

B. Differential Evolution

Differential Evolution is a population based optimization algorithm that has been largely applied to a variety of optimization problems, for instance, the calibration of hydrological models [17] and calibration of climate projections based on [18]. The algorithm works generating an initial random population of Np candidate solutions in D dimensions inside defined bounds. In each iteration, the current population is replaced by a new one via a mutation operator. The DE is presented in Algorithm 1. Its inputs are the objective function $f(\cdot)$, the population size Np , the search limits $\mathbf{b}_U, \mathbf{b}_L$, the number of dimensions D , and the number of maximum generations g_{max} . The mutation is defined in lines 8 to 15, where a new candidate solution is partially formed by a combination of three other ones or inherits several variable values from a parent. In lines 16 to 18, the best individuals are preserved and the worst are replaced to generate the next population.

Algorithm 1 DE (*best/1/bin*)

```

1: procedure DE( $f(\cdot), D, Np, \mathbf{b}_U, \mathbf{b}_L, g_{max}$ )
2:   for each  $i \in \{1 \dots Np\}$  and  $j \in \{1 \dots D\}$  do
3:      $x_{j,i} = rand_j(b_{j,U} - b_{j,L}) + b_{j,L}$ 
4:   end for
5:   for each  $g \in \{1 \dots g_{max}\}$  do
6:     for each  $i \in \{1 \dots Np\}$  do
7:        $r0, r1, r2 = \text{randomIndices}(1, Np)$ 
8:        $\mathbf{v}_i = \mathbf{x}_{r0} + F(\mathbf{x}_{r1} - \mathbf{x}_{r2})$ 
9:       for each element vector  $j \in \{1 \dots D\}$  do
10:        if  $rand_j \leq Cr$  or  $j = j_{rand}$  then
11:           $u_{j,i} = v_{j,i}$ 
12:        else
13:           $u_{j,i} = x_{j,i}$ 
14:        end if
15:      end for
16:      if ( $f(\mathbf{u}_i) \leq f(\mathbf{x}_i)$ ) then
17:         $\mathbf{x}_i = \mathbf{u}_i$ 
18:      end if
19:    end for
20:  end for
21: end procedure

```

C. EDA for a Low Number of Function Evaluations

In this article, we propose a novel Estimation of Distribution Algorithm for a Low Number of Function Evaluations (EDALNFE), shown in Algorithm 2. The optimization variables are in the domain $[0,1]$. Notice that they can be scaled and translated in the evaluation. First, a random uniform population of size $Np = 8D + 2$ is generated, where D is the number of optimization variables. Then, they are evaluated by means of Equation (7). The selection operator uses the Empirical Selection Distribution [19] to assign a probability value to each individual according to its sorting by objective function value.

The selection algorithm assigns a probability value to each of the $N = |F|$ individuals. The best individual has a value of N/Z , the second best $(N-1)/Z$, and so on, where Z is the normalization constant. The indexes of the sorted individuals are stored in $ibest$. The best individual is preserved in X_{elite} and F_{elite} . The probability values are used to compute the weighted mean and the variance estimators, as shown in [19]. The algorithm resets when stagnation is detected, using a counter of the number of consecutive generations the elite individual does not improve above a threshold, this is from lines 13 to 19. Then, the f_{reset} value is stored. It is the previous best objective value found before resetting, and it is used to test whether the f_{elite} improves or not between two resets, to stop the algorithm in case of no improvement. The search uses two search distributions to generate candidate solutions. One of them is computed in all the dimensions of the search space via a normal multivariate distribution, where each dimension is considered independent to each other, thus, only the diagonal of the covariance matrix is computed, stored, and used. The second is a distribution over a projection vector in a promising search direction. This vector is randomly generated and modified by applying to it random perturbations. Each time a perturbation is applied, the algorithm tests whether the

current projection vector U is better than the previous one by computing the Spearman correlation between the projection of the current population and their corresponding objective function values. The greater the correlation, the better the projection vector is. Notice that a perfect correlation means that the projection vector is in a perfect increasing/decreasing direction on the objective function. Hence, the proposal is sampling in all dimensions with the multivariate distribution, then, it shifts the samples over the direction of the projection vector. This sampling process is carried out according to Algorithm 3. For each generation the algorithm generates D new candidate solutions, in this case, 3 new candidate solutions, requiring a low number of function evaluations. In addition, the covariance matrix and variance of the distribution on the projection direction are scaled as follows: if f_{elite} improves, then the variance is increased, otherwise it is decreased. If it does not improve during a number of generations, according to line 19 in Algorithm 2, the population is reset by generating uniform solutions around the elite, preserving the best C_{resets} solutions.

V. EXPERIMENTS AND RESULTS

To test the approach, we optimized the PGA parameters Θ given two maps of years 2011 and 2016, respectively belonging to wake County U.S.A¹. We organized the Wake County region in training and test sub-regions as described in Fig. 1, where each cell represents a squared area of $30m^2$.

To produce the training and test sub-regions we proceeded as follows: first, we partitioned the map into 10 random Voronoi polygons. Then, seven Voronoi polygons were randomly selected and labeled as training regions. The remaining three polygons were labeled and used as test regions and used to produce PGA simulations to compare them with the ground truth.

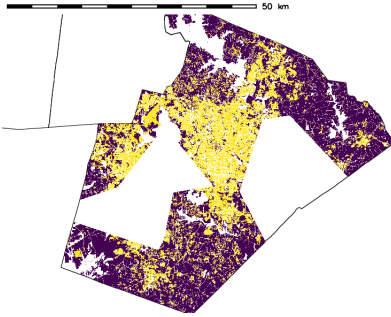


Fig. 1. Training region of Wake County U.S. in 2011. Yellow and dark color represents the developed and developable area respectively. The white region is the non developable region. The two white polygonal regions are reserved to test the PGA simulation

The training region from 2011 contains 436,314 urban cells with an increment of 21,020 in 2016. Similarly, the test region from 2011 contains 264,457 cells with an increment of 11,876

¹These maps can be downloaded from: https://github.com/ncsu-landscape-dynamics/GRASS_FUTURES/releases/tag/v2.0.0

Algorithm 2 EDALNFE

```

1: procedure EDALNFE( $f(\cdot), D$ )
2:    $Np = 8D + 2$ 
3:    $C_{elite} = 0; C_{resets} = 0$ 
4:    $X = \text{initialize}(D, Np)$ 
5:    $F = \text{evaluate}(X, f(\cdot))$ 
6:    $[ibest, p_S] = \text{selection}(F)$ 
7:    $[X_{elite}, F_{elite}] = [X_{ibest(1)}, F_{ibest(1)}]$ 
8:    $f_{reset} = F_{elite(1)}$ 
9:    $U = \text{rand}(D)$ 
10:  while  $var_c > var_{min}$  do
11:     $[\hat{X}, U] = \text{genNewInd}(X, F, X_{elite}, p_S, ibest, U)$ 
12:     $\hat{F} = \text{evaluate}(X, f(\cdot))$ 
13:    if  $|F_{elite} - F_{ibest(1)}| > |10^{-4} F_1|$  then
14:       $C_{elite} = 0$ 
15:       $[F_{elite}, F_{elite}] = [X_{ibest(1)}, F_{ibest(1)}]$ 
16:    else
17:       $C_{elite} = C_{elite} + 1$ 
18:    end if
19:    if  $C_{elite} = \text{integer}(15C_{resets} + 20)$  then
20:       $C_{resets} = C_{resets} + 1$ 
21:       $X_{elite} = X_{best(1..C_{resets})}$ 
22:       $F_{elite} = F_{best(1..C_{resets})}$ 
23:       $X = \text{reinitialize}(D, Np - 1, C_{resets}, X_{elite})$ 
24:       $F = \text{evaluate}(X, f(\cdot))$ 
25:       $[X, F] = [[X, X_{elite}], [F, F_{elite}]]$ 
26:       $[ibest, p_S] = \text{selection}(F)$ 
27:       $[X_{elite}, F_{elite}] = [X_{ibest(1)}, F_{ibest(1)}]$ 
28:       $U = \text{rand}(D)$ 
29:       $C_{elite} = 0$ 
30:      if  $F_{elite(1)} = f_{reset}$  or  $C_{resets} = Np/2$  then
31:        break
32:      else
33:         $f_{reset} = F_{elite(1)}$ 
34:      end if
35:    end if
36:     $[X, F] = [X(ibest), F(ibest)]$ 
37:     $[X, F] = [X(1..Np/2), F(1..Np/2)]$ 
38:     $[X, F] = [[X, \hat{X}], [F, \hat{F}]]$ 
39:  end while
40:  return  $[x_{elite}, f_{elite}]$ 
41: end procedure

```

urban cells in 2016. These developed cells were simulated with the best parameters found with the training regions.

We produced a suitability map by using the Logistic Regression model in Equation (1). The coefficients were estimated by using the *lme4* R package [20], to associate the urban land change produced in the 2011-2016 period given the information from 2011.

The urban land change is represented by using map algebra. The urban maps are binary where 1 means urban area, 0 means non-urban area, and null() means region not suitable for urbanization which includes protected and other excluded areas.

To produce two balanced classes for fitting the suitability function, we selected the 100% of cells presenting land-use change in the 2011-2016 period (21,605), and a random sample with the same number of invariant non-urban from a total of 1,707,283 cells.

The coefficients of each variable associated with each map

Algorithm 3 Generate New Individuals

```

1: procedure GENNEWIND( $X, F, x_{elite}, p_S, ibest$ )
2:   if first-call=True then
3:      $cnt=0; \sigma_1 = -1; \sigma_2 = -1; f_{elite}^{local} = 1e100$ 
4:   end if
5:   success=False
6:   if  $f_{elite}^{local} = F_{ibest(1)}$  and  $\sigma_1 = -1$  then
7:     success=True
8:   end if
9:    $f_{elite}^{local} = F_{ibest(1)}$ 
10:   $\mu_X = \text{weightedColumnMean}(X, p)$ 
11:   $\sigma_X = \text{weightedColumnsSD}(X, \mu_X, p)$ 
12:  if success=False then
13:     $cnt = cnt + 1$ 
14:  end if
15:  increment=True
16:  if  $cnt \geq 6$  or  $cnt = 0$  then
17:     $\sigma_X = 1.05\sigma_X$ 
18:     $Scale_1 = \max(\sigma_1/|\sigma_X|, 1)$ 
19:    if  $cnt = 10$  then
20:       $cnt = 0$ 
21:    end if
22:  else
23:     $\sigma_X = 0.6\sigma_X$ 
24:     $Scale_1 = \min(\sigma_1/|\sigma_X|, 1)$ 
25:    increment=False
26:  end if
27:   $\sigma_1 = Scale_1|\sigma_X|$ 
28:   $U_e = U$ 
29:   $corr_{max} = \text{SpearmanCorr}(XU_t, F)$ 
30:  for  $trial = 1..2000$  do
31:     $U_t = U_e + \text{randn}(D)/(4\sqrt{D})$ 
32:     $U_t = U_t/|U_t|$ 
33:     $corr_{trial} = \text{SpearmanCorr}(XU_t, F)$ 
34:    if  $|corr_{trial}| > |corr_{max}|$  then
35:       $corr_{max} = \text{sign}(corr_{trial}) |corr_{trial}|$ 
36:       $U_e = \text{sign}(corr_{trial}) U_t$ 
37:    end if
38:  end for
39:  for  $i=1..D$  do
40:     $\hat{X}_i = X_{ibest(1)} + Scale_1 \cdot \text{randn}(D) \circ \sigma_X$ 
41:  end for
42:   $U = (0.05U_e + 0.95U)/|0.05U_e + 0.95U|$ 
43:   $\sigma_t = \text{dot}(X_{ibest(1)} - X_{ibest(Np/2)}, U_e)$ 
44:  if increment=True then
45:     $Scale_2 = \max(\sigma_2/\sigma_t, 1)$ 
46:  else
47:     $Scale_2 = \min(\sigma_2/\sigma_t, 1)$ 
48:  end if
49:   $\sigma_2 = Scale_2\sigma_t$ 
50:   $factor_s = 0.6 - 2(corr_{max} - 1)^2$ 
51:  for  $i=1..D$  do
52:     $\hat{X}_i = \hat{X}_i + (factor_s \text{randn}() + factor_s)\sigma_t U$ 
53:     $\text{boundToLimits}(\hat{X}_i)$ 
54:  end for
55: end procedure

```

Coeff.	Values	Map (variables x)	Base Map
a	0.19	Intercept	N/A
β_0	0.03	Development Pressure	Urbanization 2011
β_1	0.00	Distance to interchanges	Roads in 2011
β_2	-0.48	Distance to protected areas (in kms)	Protected Areas
β_3	0.21	Dist to water shapes	Water Shape
β_4	-0.01	Smoothed map of forest	Forest
β_5	-700.33	Road Density	Roads in 2011
β_6	0.03	Travel time to city center	Roads in 2011
β_7	-0.02	Slope	Elevation

TABLE I
MAPS AND PARAMETERS FOR LOGISTIC REGRESSION

are presented in Table V.

The pressure map p'_i is computed with the following equation:

$$p'_i = \sum_{k=1}^{n_i} \frac{\text{STATE}}{d_{ik}^\gamma}, \quad (10)$$

where $\text{STATE} \in \{1, 0\}$, represents the developed or undeveloped state of cell k in the neighborhood i . This equation models the assumption that cells near to the developed center are more likely to be developed and it is controlled with γ , which in this case is set to 0.5. The suitability map is presented in Fig. 2.

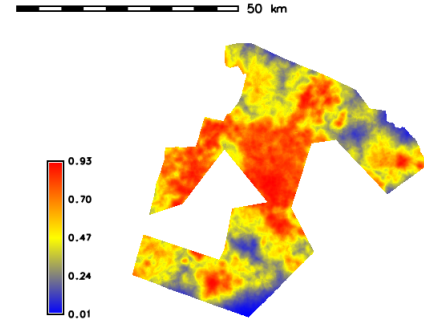


Fig. 2. Suitability map computed with the 21605 developed cells from 2011 to 2016 in the training region and 21605 random cells from the non developed cells

We compared the performance of the grid parameter identification approach implemented in the official FUTURES framework against both Differential Evolution (DE) and EDALNFE. The three approaches were limited to a maximum of 512 evaluations.

For the grid approach, every parameter was constrained to sets with 8 values evenly distributed from 0 to 1, with the exception of the discount factor φ , since 0 means no patch histogram distributions. The discrete sets of parameters are presented in Equation (11).

$$\begin{aligned}
 \mu_c &= \{0.0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1\} \\
 \sigma_c &= \{0.0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1\} \\
 \varphi &= \{0.0001, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1\}
 \end{aligned} \quad (11)$$

Therefore, we generated $512 = |\mu_c| \times |\sigma_c| \times |\varphi|$ different combinations of parameters to evaluate.

For EDALNFE, the parameter search space was constrained by the intervals $0.01 \leq \mu_c, \sigma_c \leq 1$, and $0.0001 \leq \varphi \leq 1$.

For DE, the parameter search space was constrained by the intervals $0 \leq \mu_c, \sigma_c \leq 1$, and $0.0001 \leq \varphi \leq 1$.

The DE settings are those given by default in the scikit-learn implementation [21], but the population size was set to 16 with a maximum of 31 iterations to limit the optimization process to 512 evaluations with polish parameter to false to avoid the use of the L-BFGS-B optimization method after DE finishes.

We take as reference the Grid approach result where the best combined error found was 22.64 using 512 different evaluations. The number of evaluations are the 100% of the total time assigned to each optimization algorithm to optimize. The EDALNFE and DE were executed 5 independent times and the results are reported in Table II.

run	EDALNFE	#Eval.	Time	DE	#Eval.	Time
1	28.95	92	17.97%	18.43	512	100%
2	20.19	119	23.93%	18.59	512	100%
3	20.74	100	19.53%	18.7	512	100%
4	18.97	131	25.59%	18.48	512	100%
5	18.22	512	100.00%	18.64	512	100%

TABLE II

RESULTS OF 5 INDEPENDENT RUNS OF EDALNFE AND DE

For EDALNFE, we observed that for 4 out of 5 runs it obtained better results than Grid search with the lowest number of evaluations, but the first run with the worst result (combined error in bold) prematurely converged. The fifth execution presented the best combined error from all the executions. In the case of DE, 5 out of 5 runs obtained better results than the Grid search, stopping when 512 evaluations were reached.

A. PGA Simulations With the Best Experimental Result

Once we found the best PGA parameters, we proceeded to simulate the urban land change with those parameters. The best combined error of 18.22 was obtained with the fifth execution of EDALNFE $\varphi = 0.2663$, $\mu_c = 0.113$ and $\sigma_c = 0.998$.

Since every PGA execution produced a different scenario, we ran 100 PGA independent simulations by using the test sub-region hoping to observe a consistent PGA pattern.

In Fig. 3 we present the map of a frequency histogram in the geographical space, where darkest green areas are those where PGA simulated urban development more times. The orange cells represent the actual development observed from 2011 to 2016. From all the independent runs, the maximum number of times that a cell was urbanized was 20 out of 100.

In order to appreciate with more detail the PGA performance in the test regions, we zoom-in an arbitrary small region in Fig. 4 to observe with more clarity the PGA pattern.

In this map, the white cells are excluded areas not suitable for urbanization (e.g. they were developed in 2011, represents a water shape, or a protected region). The light green cells are suitable for development and darkest green cells are the cells with more simulated urban development according to PGA. The darkest cells show that 20 simulations generated urban



Fig. 3. Test region of PGA in year 2016

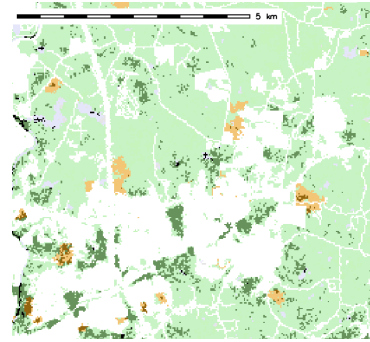


Fig. 4. Heat map of 100 runs. The Darkest regions where urbanized more times than light regions

developing and the lightest green cells show no urban land change in any simulation.

The orange regions are the actual developed regions used as ground truth for comparison purposes.

When contrasting the green dark region with the ground truth, we observed that most of the developed regions included at least a dark green cell. In the map, dark green areas represent approximately the same probability of being urbanized. These probabilities do not pretend to indicate exactly where urbanization will occur, but the level of uncertainty of the urban development process.

VI. CONCLUSIONS

We presented the formulation of PGA calibration as an optimization problem. To the best of our knowledge, it is the first time that PGA is automatically calibrated with an optimization algorithm and contrasted to grid search.

We introduce the Estimation of Distribution Algorithm for a Low Number of Function Evaluations (EDALNFE), that was designed for the calibration task.

EDALNFE is a competitive optimization algorithm that does not require expert settings and addresses the generalized problem in Evolutionary Algorithms (EAs) of defining a stopping criterion. Most of the EAs arbitrarily stop when a number of generations or function evaluations is reached. Other implementations additionally used some criteria based on a lack of improvement of the elite individual. The results of the EDALNFE show that it is competitive against the DE and the grid search for the PGA calibration, highlighting the low number of the evaluations used to find the best results.

Nevertheless, none of these criteria actually measures convergence. The EDALNFE converge in distribution. The stopping criterion uses the variance of the objective function. For a continuous function, the convergence in the image ensures the convergence in the search space, but measuring the convergence in the objective function space (or image), is computationally cheaper than measuring it in the search space.

The EDALNFE delivers competitive results for this calibration task, thus, likely it would be competitive in other similar problems.

Hence, future work contemplate the application of EDALNFE to other calibration tasks.

In addition, in this work we measure the performance of the prediction using distances between histograms. Nonetheless, they are used in this work since they are recommended by the FUTURES' authors and because introducing a particular objective function for a novel algorithm could lead to a biased conclusion about the performance. Therefore, it is fair to use a standard objective function with a novel algorithm to ensure an unbiased comparison. In this same regard, these measures do not quantify the actual prediction. Thus, future work also contemplates proposing a suitable performance measure that actually quantifies the prediction of the urban land, considering the urban structure of the geographic space accompanied with more experiments to analyze statistically the performance of the optimization algorithms.

According to the experimental conclusions, we found that the size and shape patch distribution distances are not enough for assessing the PGA. It is necessary to consider and study in a deeper way at least two additional aspects:

- 1) To design a better urban-structure characterization that does not depend on the circle shape, but considers the diversity of the patch shapes. As a consequence it would be necessary to adapt the PGA or to create a new urban simulation model that considers the complex characteristics of the urban structure to produce more realistic urban scenarios.
- 2) To consider a metric or a distance that takes into account the mismatch of the land use prediction directly in the geographic space.

The design of an urban simulation model should be accompanied with a calibration methodology that benefits the applicability of the model. We also think that the PGA distribution maps can be generated using conditional probabilistic models that produce similar results and could potentially substitute PGA simulations, reducing the computational cost.

ACKNOWLEDGMENTS

S. Ivvan Valdez, Rodrigo Lopez-Farias, and Alberto Garcia-Robledo are supported by Cátedra-CONACYT 7795, 7029 and 6245 respectively. The experiments were executed in the CentroGeo cloud computing project "Nimbus".

REFERENCES

- [1] B. Pickard, J. Gray, and R. Meentemeyer, "Comparing quantity, allocation and configuration accuracy of multiple land change models," *Land*, vol. 6, no. 3, p. 52, 2017.
- [2] S. H. Clarke, Keith C. and L. Gaydos, "Methods and techniques for rigorous calibration of a cellular automaton model of urban growth," in *Proceedings of the Third International Conference/Workshop on Integrating GIS and Environmental Modeling*, 1996.
- [3] "r.futures calib - GRASS GIS manual," <https://grass.osgeo.org/grass78/manuals/addons/r.futures.calib.html>, accessed: 2021-06-02.
- [4] W. Veerbeek, A. Pathirana, R. Ashley, and C. Zevenbergen, "Enhancing the calibration of an urban growth model using a memetic algorithm," *Computers, Environment and Urban Systems*, vol. 50, pp. 53–65, 2015.
- [5] J. Shan, S. Alkheder, and J. Wang, "Genetic algorithms for the calibration of cellular automata urban growth modeling," *Photogrammetric Engineering & Remote Sensing*, vol. 74, no. 10, pp. 1267–1277, 2008.
- [6] J. Jafarnejad, A. Salmanmahiny, and Y. Sakieh, "Subjectivity versus objectivity: comparative study between brute force method and genetic algorithm for calibrating the sleuth urban growth model," *Journal of Urban Planning and Development*, vol. 142, no. 3, p. 05015015, 2016.
- [7] F. Naghibi, M. R. Delavar, and B. Pijanowski, "Urban growth modeling using cellular automata with multi-temporal remote sensing images calibrated by the artificial bee colony optimization algorithm," *Sensors*, vol. 16, no. 12, p. 2122, 2016.
- [8] J. Yang, J. Gong, W. Tang, Y. Shen, C. Liu, and J. Gao, "Delineation of urban growth boundaries using a patch-based cellular automata model under multiple spatial and socio-economic scenarios," *Sustainability*, vol. 11, no. 21, p. 6159, 2019.
- [9] N. N. Pinto and A. P. Antunes, "A cellular automata model based on irregular cells: application to small urban areas," *Environment and planning B: planning and design*, vol. 37, no. 6, pp. 1095–1114, 2010.
- [10] K. Al-Ahmadi, L. See, A. Heppenstall, and J. Hogg, "Calibration of a fuzzy cellular automata model of urban dynamics in saudi arabia," *Ecological complexity*, vol. 6, no. 2, pp. 80–101, 2009.
- [11] A. García, I. Santé, M. Boullón, and R. Crecente, "Calibration of an urban cellular automaton model by using statistical techniques and a genetic algorithm. application to a small urban settlement of nw spain," *International Journal of Geographical Information Science*, vol. 27, no. 8, pp. 1593–1611, 2013.
- [12] M. G. Turner, R. H. Gardner, R. V. O'Neill, and R. V. O'Neill, *Landscape ecology in theory and practice*. Springer, 2001, vol. 401.
- [13] M. Janga Reddy and D. Nagesh Kumar, "Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review," *H2Open Journal*, vol. 3, no. 1, pp. 135–188, 2021.
- [14] S. Nandi and M. Janga Reddy, *Parameter Estimation of a Macroscopic Hydrological Model Using an Adaptive Differential Evolution*. Cham: Springer International Publishing, 2021, pp. 243–255. [Online]. Available: https://doi.org/10.1007/978-3-030-58051-3_17
- [15] T. V. Dinh, H. Nguyen, X.-L. Tran, and N.-D. Hoang, "Predicting rainfall-induced soil erosion based on a hybridization of adaptive differential evolution and support vector machine classification," *Mathematical Problems in Engineering*, vol. 2021, 2021.
- [16] L. Blanco-Cocom, S. Botello-Rionda, L. Ordoñez, and S. I. Valdez, "Robust parameter estimation of a pemfc via optimization based on probabilistic model building," *Mathematics and Computers in Simulation*, vol. 185, pp. 218–237, 2021.
- [17] O. Adeyeri, P. Laux, J. Arnault, A. Lawin, and H. Kunstmann, "Conceptual hydrological model calibration using multi-objective optimization techniques over the transboundary komadugu-yobe basin, lake chad area, west africa," *Journal of Hydrology: Regional Studies*, vol. 27, p. 100655, 2020.
- [18] Y. Hundedcha, B. Arheimer, P. Berg, R. Capell, J. Musuza, I. Pechlivaniadis, and C. Photiadou, "Effect of model calibration strategy on climate projections of hydrological indicators at a continental scale," *Climatic Change*, vol. 163, no. 3, pp. 1287–1306, 2020.
- [19] S. I. Valdez-Peña, A. Hernández-Aguirre, and S. Botello-Rionda, "Approximating the search distribution to the selection distribution in edas," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, 2009, pp. 461–468.
- [20] D. Bates, M. Mächler, B. Bolker, and S. Walker, "Fitting linear mixed-effects models using lme4," *Journal of Statistical Software*, vol. 67, no. 1, pp. 1–48, 2015.
- [21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.