MUSE ArcGIS Pro Toolbox User's Manual

2023/11

China University of Geosciences (Wuhan)

https://www.landinsightlab.com/

contents

1.Introduction 1
2.Operation Environment........ 1
3.Data Preparation 1
3.1 CSV documents 2
3.1.1 Stepwise demand of urban development data 2
3.1.2 History patch size controller data 2
3.1.3 Gaussian parameters data 3
3.1.3 Stepwise percent of organic growth data 3
3.2 TIF files 4
4. User Guide for MUSE Toolbox 5
4.1 Toolbox Installation 5
4.2 Running the Toolbox........ 7
4.2.1 Model Validation 8
4.2.2 Scenario simulation 12
4.3 Explanation of simulation results 13
5. The principles of model........ 14
5.1 Basic model framework........ 14
5.2 Controlling the extent of urban expansion........ 15
5.3 Patch size controller 15
5.3.1 Log-normal distribution patch size controller 15
5.3.2 Power law distribution size controller 16
5.3.3 Historical period patch size controller........ 16
5.4 Patch generation engine 16
5.4.1 SPGE........ 16
5.4.2 PPGE........ 18
5.4.3 Nei-PGE 20
5.4.4 Dis-PGE........ 20
5.5 Explanation of Important Model Parameters 21
5.5.1 Proportion of organic growth patches........ 21
5.5.2 Uncertainty of patch location 21
5.5.3 Pruning coefficient for patch seed library 22
6. Copyright statement and contact information 23

1.Introduction

The MUSE (Multi-engine Urban Expansion Simulator) is a cellular automata-based model for simulating urban expansion, incorporating four patch generation algorithms and three patch size generators. Its primary objective is to replicate the process and pattern of urban land expansion. The four patch generation algorithms are as follows:：

patch-growing engine with distance-constrained patch shape (Dis-PGE);

patch-growing engine with Neighborhood-constrained patch shape (Nei-PGE);

simple patch-growing with maximized cell suitability (SPGE);

parameterized patch-growing engine with tradeoff between maximized cell suitability and optimized patch shape (PPGE).

In addition, MUSE provides three patch size controllers to determine the size of patches:

Lognormal Distribution：Patch Size Controller assumes that the patch area follows a log-normal distribution (refer to Equation (4)).

Power Distribution：The Patch Size Controller assumes that the patch area follows a power-law distribution (refer to Equation (5)).

History：The third type of Patch Size Controller directly utilizes historical patch area sizes from previous periods.

By utilizing MUSE, users are able to simulate the process of urban expansion, generate patches of varying shapes and sizes through different algorithms and generator choices, as well as produce diverse patterns of spatial structure for urban land. This model holds significant application value in the fields of urban planning, land use planning, land resource management and decision support.

2.Operation Environment

Items	System Requirements
Operating System	Windows 10 and above
CPU	Intel® Core™ i5 7th generation or newer / AMD Ryzen™ 5 2nd generation or newer
Memory	8GB of RAM or more
Storage Space	At least 10GB of available space
Software Environment	ArcGIS Pro 2.7 or later

3.Data Preparation

When preparing the data for MUSE, the first step is to locate the "_TEST_FILES" folder in the MUSE toolbox, for example: C:\Users\Administrator\Desktop\MUSE_FOR_ArcGIS\

_TEST_FILES. This folder contains 9 files used for examples, specifically covering 4 files in CSV format and 5 files in TIFF format. These files are designed to be the necessary input data for running MUSE. The next steps will revolve around these example files, providing detailed explanations of the data formats required for MUSE operation.

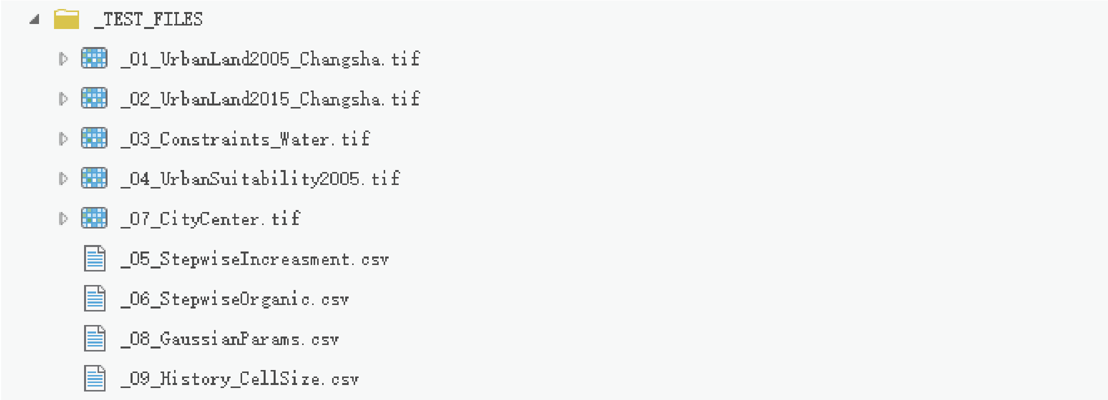

Figure 3-1: Sample Files Directory

3.1 CSV documents

3.1.1 Stepwise demand of urban development data

The stepwise demand of urban development data is stored in CSV format, consisting of two columns. The first column represents various simulation time points, where the time unit in this example is years (but it could be any time unit). The second column represents the corresponding increment of urban land grid quantity at each time point. These grid quantities reflect the growth area of urban land in the study area for each time period (the time interval between consecutive time points), measured in grid units. The example file "_05_StepwiseIncreasment.csv" records the annual increment of urban land grid quantity in the study area from 2005 to 2015.

	A	B	C	D	E	F	G
1	year	cellNum					
2	urban_area_2006	15403					
3	urban_area_2007	18573					
4	urban_area_2008	20348					
5	urban_area_2009	23048					
6	urban_area_2010	25043					
7	urban_area_2011	27583					
8	urban_area_2012	29746					
9	urban_area_2013	39325					
10	urban_area_2014	43048					
11	urban_area_2015	57355					

Figure 3-2: Stepwise demand file for urban development

3.1.2 History patch size controller data

The History Patch Size Controller data is stored in CSV format, with each row containing two columns. The first column represents the patch number for historical periods, and the second column represents the corresponding patch area size. Patch sizes are recorded as integers in the file, where each row of data indicates the number of raster units included in the patch. The example file "_09_History_CellSize.csv" records data for 5000 patch sizes, as illustrated in Figure 3-3.

	A	B	C
1	**id**	**patchSize**	
2	1	323.91583	
3	2	210.3503525	
4	3	285.3129673	
5	4	411.7282632	
6	5	491.2518693	
7	6	463.554796	
8	7	227.6863918	
9	8	426.3618885	
10	9	329.4061098	
11	10	280.9442427	

Figure 3-3: Example of patch sizes in historical periods

3.1.3 Gaussian parameters data

The Gaussian Parameters data is stored in CSV format, containing three columns. The first column represents various simulation time points. The second and third columns contain the values of the mean parameter b (in units of kilometers) and the standard deviation parameter c (in units of kilometers) for the Gaussian function, respectively. These values can be in decimal or integer form. The example file "_08_GaussianParams.csv" records the values of the two parameters b and c for the Gaussian correction file during the time period from 2005 to 2015, as shown in Figure 3-4.

	A	B	C
1	**year**	**b(/km)**	**c(/km)**
2	2006	10	5
3	2007	10	5
4	2008	10	5
5	2009	10	5
6	2010	10	5
7	2011	10	5
8	2012	10	5
9	2013	10	5
10	2014	10	5
11	2015	10	5

Figure 3-4: Example file of gaussian parameters

3.1.3 Stepwise percent of organic growth data

The stepwise percent of organic growth data is stored in CSV format, containing two columns. The first column represents various simulation time points, and the second column represents the corresponding organic patch area ratio at each time point. This ratio is stored as a decimal between 0 and 1. It's worth noting that the area ratio of spontaneously growing patch types is equal to 1 minus the organic patch area ratio. The example file "_06_StepwiseOrganic.csv" records the increment of organic patch area ratios from 2005 to 2015, as illustrated in Figure 3-5.

	A	B	C	D	E	F	G
1	year	organic					
2	urban_area_2006	0.7					
3	urban_area_2007	0.75					
4	urban_area_2008	0.77					
5	urban_area_2009	0.75					
6	urban_area_2010	0.73					
7	urban_area_2011	0.71					
8	urban_area_2012	0.72					
9	urban_area_2013	0.7					
10	urban_area_2014	0.77					
11	urban_area_2015	0.79					

Figure 3-5: Example File of Stepwise percent of organic growth data

3.2 TIF files

Users are required to provide five TIFF files, encompassing data on the spatial distribution of urban construction land at the simulation's base period, spatial distribution data at the model verification period, a file indicating the probability of urban construction suitability, a file specifying constraints on urban development and construction, and data on urban center points. These files must strictly adhere to consistency in spatial extent, including the same number of rows and columns, projection coordinates, and spatial resolution. Example files are illustrated in Figure 3-6.

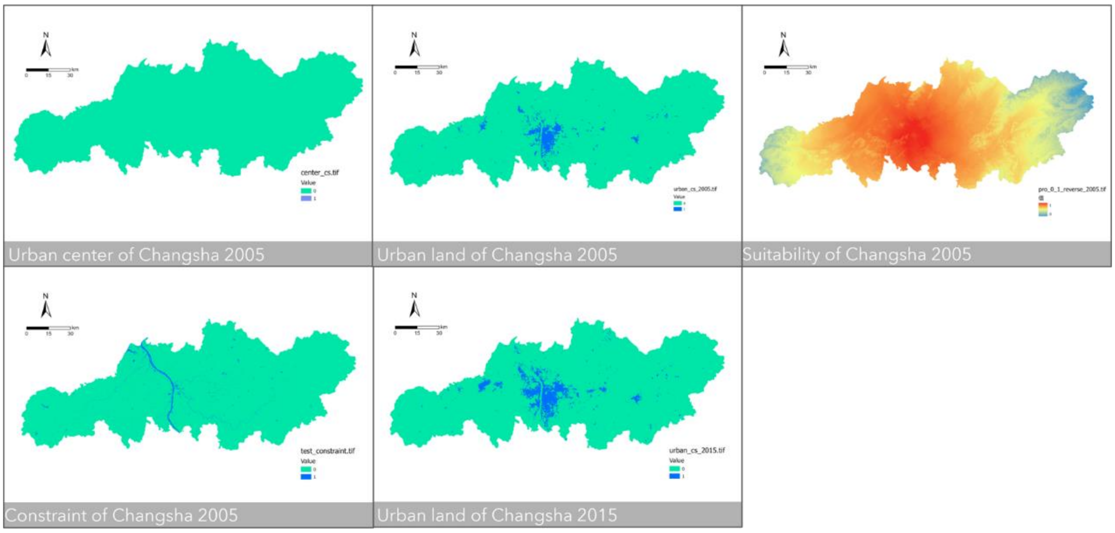

Figure 3-6: Model input data files

(1) The urban construction land data of the base period depicts the spatial distribution of urban construction land during that time. The values in this dataset are binary, with 0 representing undeveloped land (i.e., non-urban construction land) and 1 indicating developed land (i.e., urban construction land).

(2) The spatial distribution data of urban construction land at the model verification point reflects the distribution of urban construction land at that specific location for validation purposes. Similar to the urban construction land data during the base period, each grid unit in this dataset has a value range between 0 and 1, where 0 represents undeveloped land and 1 represents developed land. This dataset is used to compare with the urban construction land

distribution data generated by the MUSE model at this specific time point, thereby evaluating the simulation accuracy of the model.

(3) The urban construction suitability probability file is the outcome of an assessment on the suitability of urban development based on various driving factor data. This document provides probabilistic information regarding the appropriateness of urban construction and serves as crucial data for MUSE model simulation. The values in the urban construction suitability probability data range from 0 to 1, with higher values indicating a greater degree of suitability for urban development.

(4) The document on Urban Development and Construction Restrictions provides information regarding the spatial extent of restricted urban development. It records areas designated for urban land development, including water bodies, permanent basic farmland, historical and cultural reserves, based on restrictions set by the user in accordance with simulation scenario requirements. The values in this file are represented as 0 or 1, where 0 indicates developable areas and 1 indicates restricted development zones.

(5) The urban central point data defines the location of the urban central point, typically chosen within the city's central business district (CBD), government agency locations, or important city hubs. This raster data has a value range of 0 to 1, where 0 represents non-central grid points and 1 represents central grid points. Each center is represented by a grid, so the number of grids with a value of 1 corresponds to the number of urban centers.

Table 1: List of model input data files

File Names	Data Sources	Values Range	Corresponding Example File
Simulated Base-Year Urban Construction Land Spatial Distribution Data	Remote Sensing Data, Land Survey Data, etc.	0 (Undeveloped Land), 1 (Developed Land)	_01_UrbanLand 2005_Changsha .tif
Model Validation Urban Construction Land Spatial Distribution Data	Remote Sensing Data, Land Survey Data, etc.	Remote Sensing Data, Land Survey Data, etc.	_02_UrbanLand 2015_Changsha .tif
Urban Construction Suitability Probability File	Evaluated based on Driver Factor Data	0-1 (Urban Construction Suitability Probability)	_04_UrbanSuita bility2005 .tif
Urban Development and Construction Restriction File	Set based on Real Conditions and Simulated Scenario Requirements	0 (Developable Area), 1 (Restricted Development Area)	_03_Constraints _Water .tif
Urban Center Point Data	Set based on Real Conditions and Simulated Scenario Requirements	0 (Non-Center Point), 1 (Center Point)	_07_CityCenter . tif

4. User Guide for MUSE Toolbox

4.1 Toolbox Installation

(1) Acquiring the Python version of ArcGIS Pro involves the initial retrieval and extraction of the compressed MUSE toolbox file to a designated directory. Subsequently, within the ArcGIS

Pro interface, proceed through the “Project” and “Package Manager” tabs, selecting “python” to ascertain the current Python version underpinning ArcGIS Pro. As illustrated in Figure 4-1, the provided example denotes the Python version as 3.9.11. The ensuing procedural example will be delineated using Python 3.9 as the reference version.

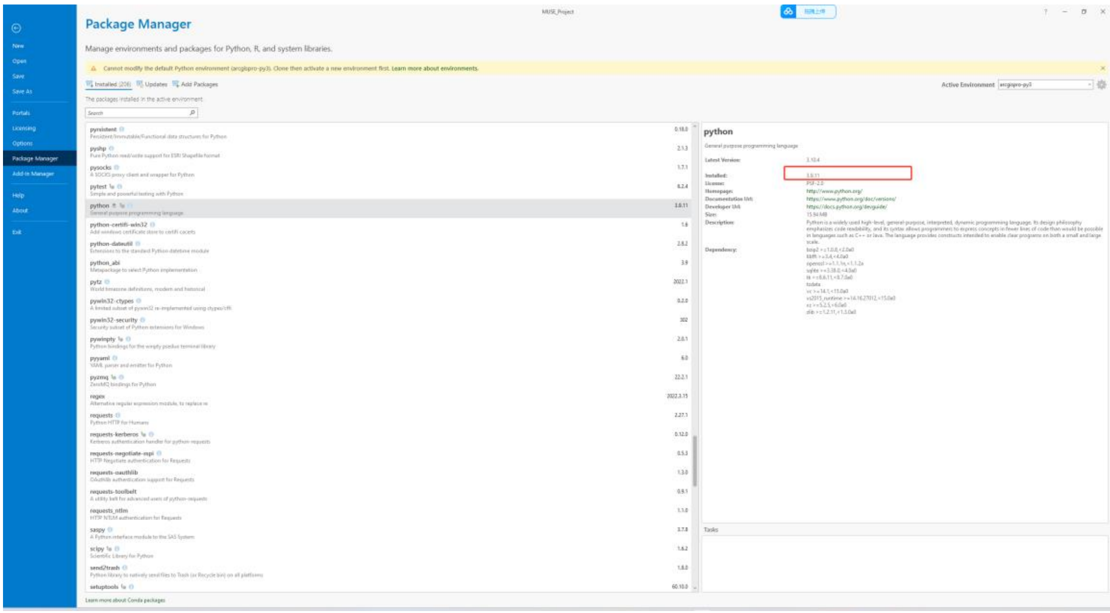

Figure 4-1 ArcGIS Pro python version

(2) To replace the corresponding MUSE_CMD file for the Python version, navigate to the “MUSE_CMD” folder. Locate the MUSE_CMD file that corresponds to the python3.9 version and copy it to the “MUSE_Backend” folder. During this process, delete the original version of the MUSE_CMD file to ensure that only one MUSE_CMD file is retained in the “MUSE_Backend” folder. It is crucial to emphasize that all files within the “MUSE_Backend” directory are essential for the proper operation of the MUSE toolbox. Therefore, this directory must be in the same root directory as the MUSE_Script.py script file, and the directory name should not be altered. Within the “MUSE_CMD” directory, the “cp36” in the file name indicates compatibility with ArcGIS Pro Python 3.6 version, and so forth. For example, the file MUSE_CMD.cp311-win_amd64.pyd corresponds to the version suitable for ArcGIS Pro Python 3.11.

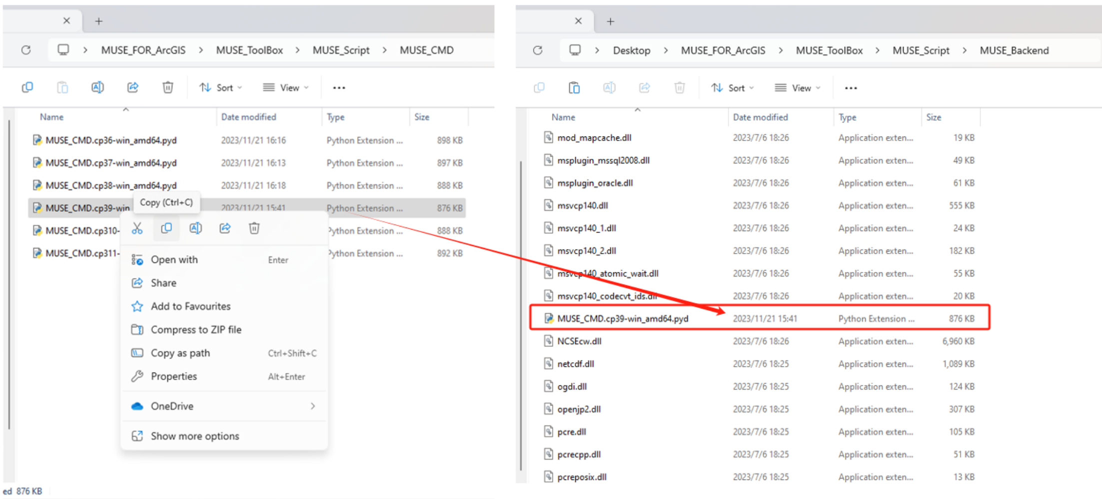

Figure 4-2 Replace the MUSE_CMD file

(3) To import the toolbox in ArcGIS Pro, first, in the directory pane, locate “Project” > “Toolboxes”. Right-click and select “Add Toolbox”, then navigate to the directory where the MUSE toolbox is saved. In that directory, find and open the “MUSE.tbx” file, and finally, click “OK”, as illustrated in Figure 4-3. Once the toolbox is added, you will observe two tools: “MUSE_CH” and “MUSE_ENG”, corresponding to the Chinese and English versions of the MUSE toolbox. This guide will demonstrate the operation using the English version of the toolbox, and the procedure for the Chinese version is identical to the English version.

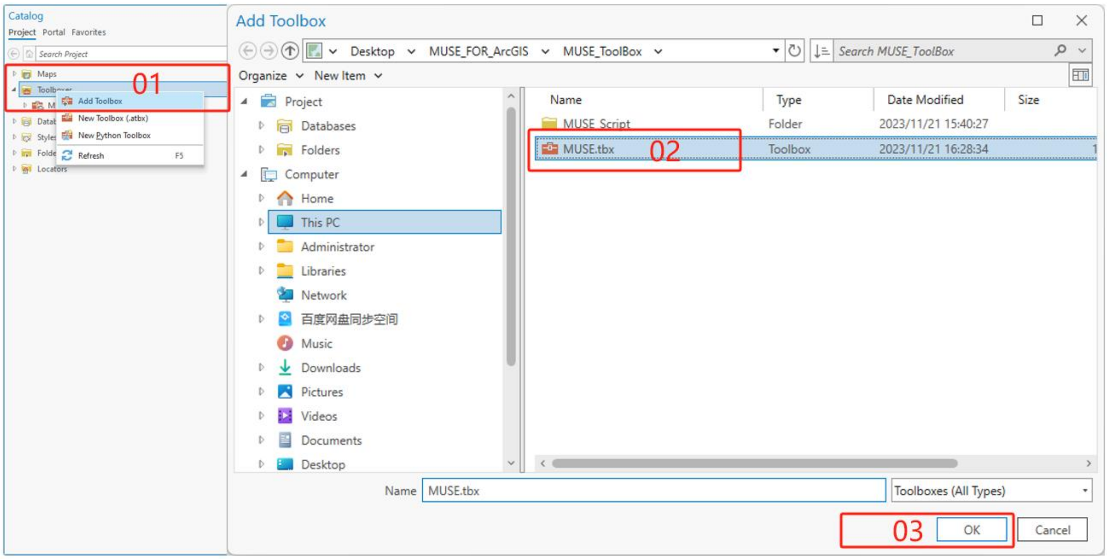

Figure 4-3 Add toolbox

4.2 Running the Toolbox

Initially, initiate the MUSE toolbox’s primary interface by double-clicking on the “MUSE_CH” tool, as depicted in Figure 4-4. Within this interface, the foremost consideration is the mode selection, where the MUSE toolbox offers two distinctive modes: model validation and scenario prediction. Subsequently, attention is directed toward the data input segment. In the model validation mode, six data elements must be provided, encompassing four raster file datasets and two CSV file datasets. Following this, the parameterization for Gaussian adjustment is undertaken. Users are then prompted to input global model parameters, including simulation start and end periods, patch position uncertainty, seed cell library pruning coefficient, and neighborhood type – comprising a set of five overarching parameters. Sequentially, controls for patch size and the choice of the patch generation engine are presented. Ultimately, a pivotal consideration is the specification of the output data address. Subsequent sections will elucidate the nuanced input methodologies for each parameter under both the model validation and scenario prediction modes.

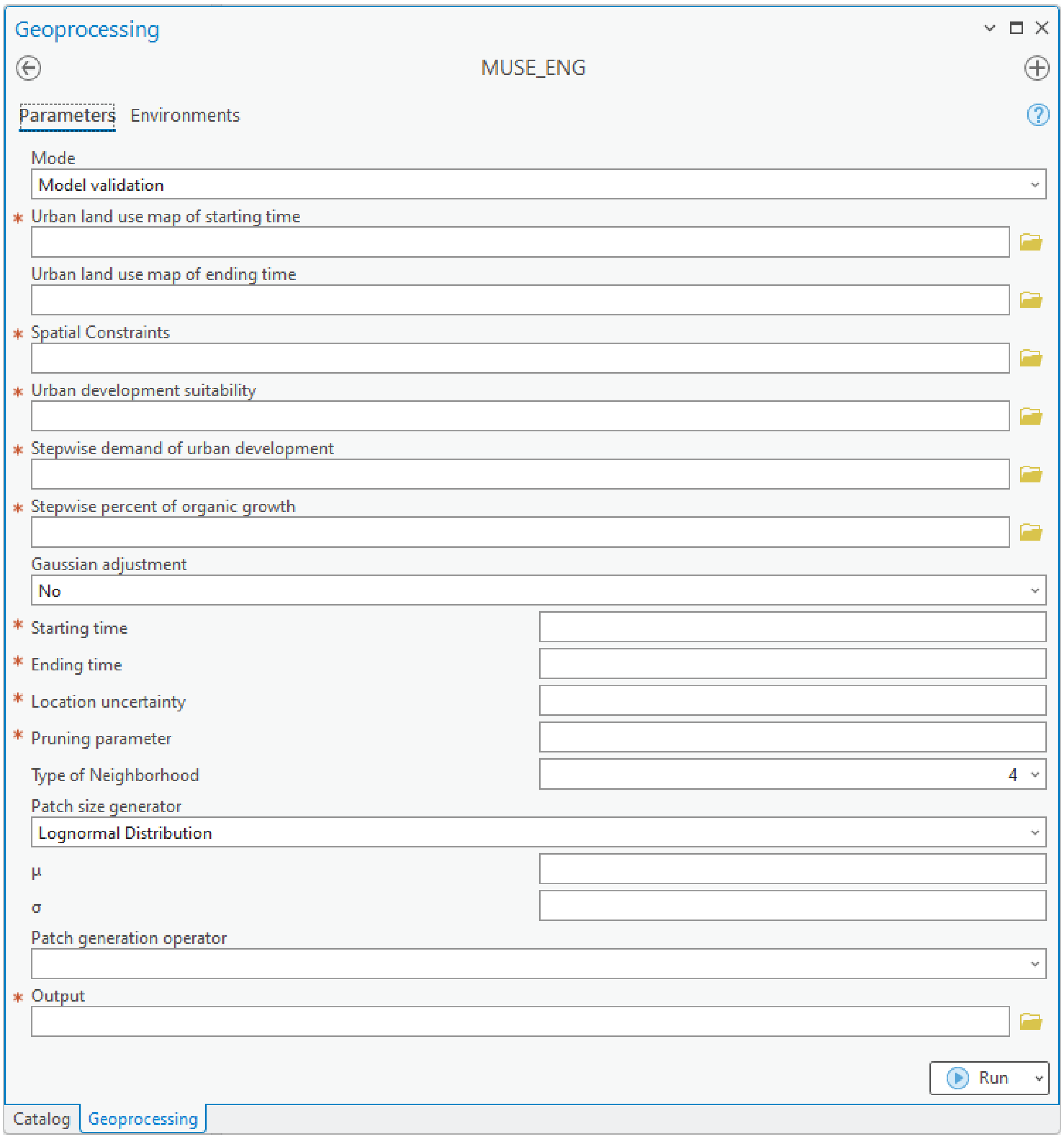

Figure 4.4 The main interface of MUSE toolbox

4.2.1 Model Validation

(1) Data Input: Sequentially input the requisite data for the MUSE tool in the first five file input fields. The corresponding relationships between the required input data and the example files in the _TEST_FILES folder are delineated in Table 2.

Table 2: Information overview of input files

Parameters Name	File Types	Example Files
Urban land use map of starting time	Base-year Urban Construction Land Data	_01_UrbanLand2005_Changsha.tif
Urban land use map if ending time	Target-year Urban Construction Land Data	_02_UrbanLand2015_Changsha.tif
Spatial Constraints	Urban Development Restriction File	_03_Constraints_Water.tif
Urban development suitability	Urban Construction Suitability Probability File	_04_UrbanSuitability2005.tif

Stepwise demand of urban development	Urban Construction Land Increment	_05_StepwiseIncreasment.csv
Stepwise percent of organic growth	Patch organic growth category proportion data	_06_StepwiseOrganic.csv

(2) Expansion Extent Control: The management of expansion extent involves the decision to deploy the Gaussian adjustment control module. If this module is selected, confirm the choice as "Yes" in the dropdown box. Subsequently, systematically input data for city center points, Gaussian correction parameters, and the weight representing the attractiveness of urban land development to the city center. Within the MUSE tool, this weight is referred to as the Gaussian function weight, constrained within the range of 0 to 1. For an in-depth understanding of the module's underlying principles, please refer to Section 5.2. The corresponding interface for this module is depicted in Figure 4-5.

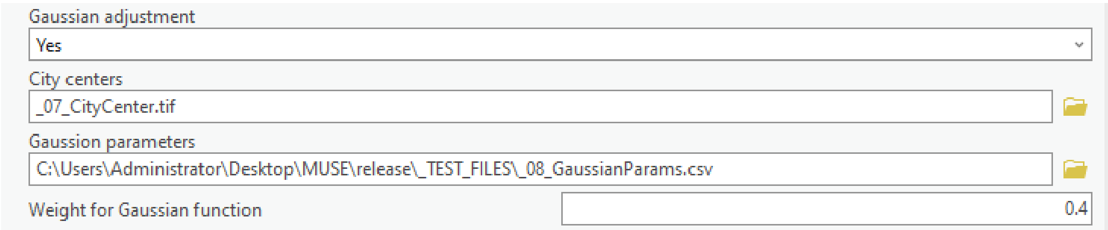

Figure 4-5 Parameter input interface of expansion degree control module

Table 3: Overview of input files information for expansion degree control module

Parameters Name	File Types	Example Files
Urban Center	City center raster data	_07_CityCenter.tif
Gaussian Parameters	Parameters data based on Gaussian correction rule	_08_GaussianParams.tif

(3) Input of Global Parameters: In the realm of global parameters for the model, precise configuration of five pivotal parameters is imperative. Notably, the total duration of the simulation period (computed as the simulation end minus the simulation start) demands meticulous attention to specific constraints. Its numerical value must not surpass the temporal span delineated in the urban construction land increment data. To elucidate, this value is determined by subtracting one from the total number of rows in the urban construction land increment data, denoting the temporal extent. For instance, if the urban construction land increment data (inclusive of the header) encompasses 11 rows, spanning incremental information from 2006 to 2015, the designated time span parameter amounts to 10 years. Consequently, when configuring the simulation period length, the value should be less than or equal to 10 years—indicating that the disparity between the simulation end and simulation start should not exceed 10 years. The signification and admissible ranges of each parameter within the global parameter section are expounded upon in the following table:

Table 4: Model parameters

Parameters Name	Parameters Description	Value Range
Starting time	Starting step of the model simulation	1~36767
Ending time	Ending step of the model simulation	1~36767

Loacation uncertainty	Proportion of non-randomly selected seeds in the seed selection process for patches	0~1
Pruning parameter	The size of the patch seed unit library is equal to the total number of developable grid units sorted in descending order based on development probability, multiplied by a pruning coefficient.	0~1
Type of neighborhood	In the context of 4-neighborhood, it corresponds to the Von Neumann neighborhood, while in the case of 8-neighborhood, it corresponds to the Moore neighborhood.	4、8

(4) Selection of Patch Size Generator: Opt for one of three available patch size generators. The "Lognormal Distribution" and "Power-law Distribution" represent intrinsic generation strategies embedded within the model, assuming distinctive distributions—lognormal and power-law, respectively—for newly generated patch sizes. Alternatively, the selection of "Historical Period Patch Sizes" mandates the provision of a bespoke CSV document delineating customized patch sizes, as comprehensively expounded in Section 3.1. The "Lognormal Distribution" strategy relies on a lognormal distribution random generator to govern the generation of patch sizes, necessitating the input of parameters such as the mean and log standard deviation of the lognormal distribution. On the other hand, the "Power-law Distribution" strategy leverages a power-law distribution random generator to regulate the generation of patch sizes, requiring input parameters encompassing the scaling constant and exponent of the power-law distribution.

(5) Selection of Patch Generation Engine: MUSE offers a choice among four algorithmic engines for patch generation. It is noteworthy that, when opting for the neighborhood control patch generation engine, the model inherently configures the patch position uncertainty parameter to 1 and the neighborhood type parameter to 8. This adjustment is a consequence of the reliance on stochastic processes and iterative neighborhood mechanisms inherent in the neighborhood control patch generation engine. A comprehensive overview of the parameters associated with each engine is elucidated in the following table:

Table 5: Explanation of engine control parameters

Explanation of Engine Control Parameters				
Engines Name	**Parameters Name**	**Parameters Description**	**Default Value**	**Value Range**
SPGE	This engine does not require any input parameters.			
PPGE	N	N and D together influence the longest dimension of the plaque	1	Greater than 0
	D		2	Greater than 0
	A	the number of arms	2	Not less than 0
	O	patch orientation	45	Not less than 0
	suit_weight	The weight of the patch shape during the generation process	0.5	0-1

	shape_weight	The weight of suitability during the generation process	0.5	0-1
Nei-PGE	beta	Whether neighborhood repetition based on seed units controls the compactness of the patch	1.6	Greater than 0
Dis-PGE	delta	Control of patch shape based on a distance decay mechanism	2	Any real number

(6) Output Location Selection: In the course of selecting output results, it is necessary to provide a filename for result preservation. Throughout this process, one has the choice to append a file extension, for instance, “Exp_CS.tif”. Alternatively, the option exists to forgo a file extension and directly input “Exp_CS” with the model autonomously completing the filename. Figure 4-6 serves as an exemplification of model parameter configurations.

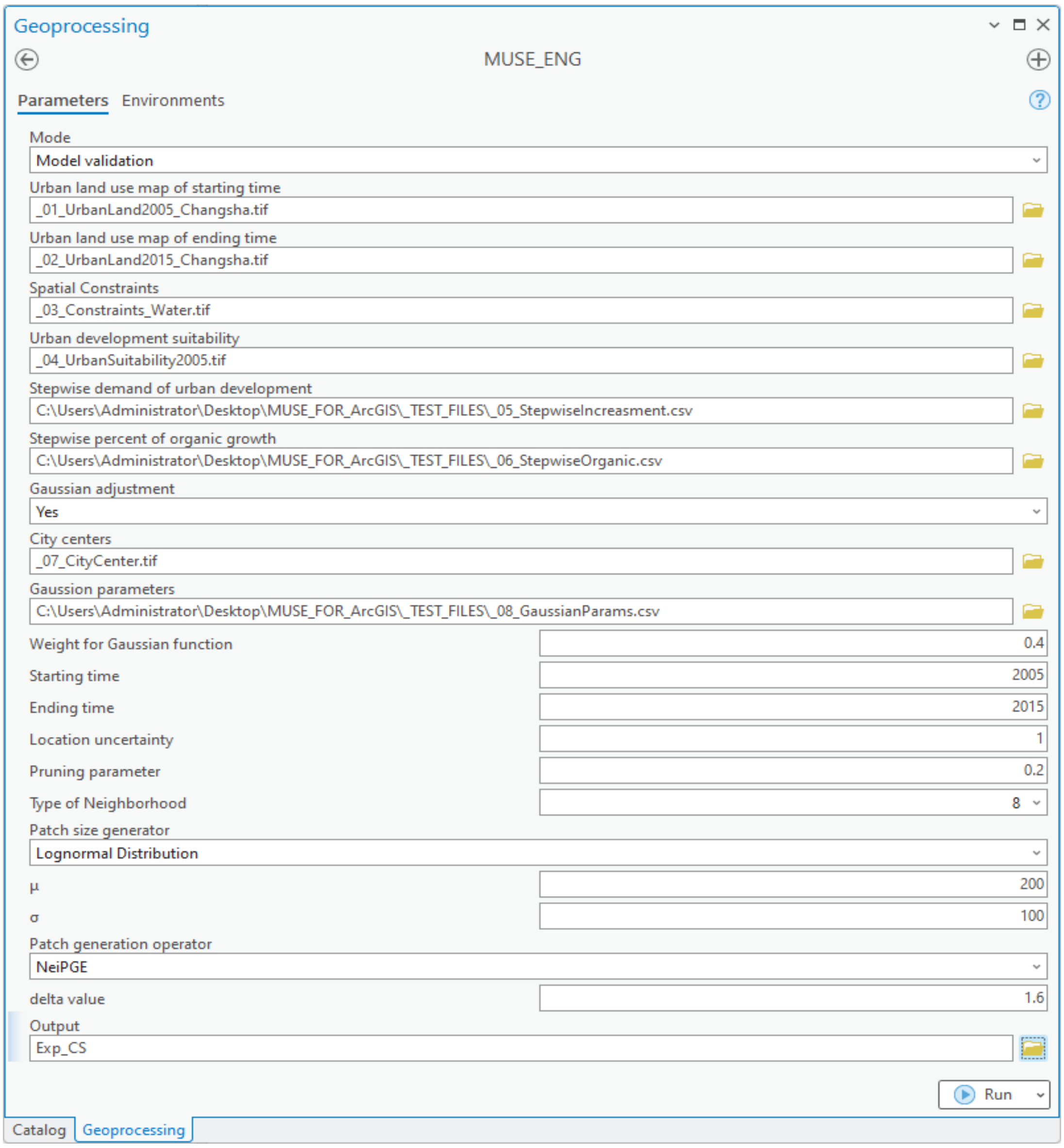

Figure 4-6 Example model parameter settings

(7) Examination of Simulation Results: Following the simulation's conclusion, the resultant files are preserved in the specified directory. Accessing pertinent information from the model output can be achieved by selecting "View Details". In reference to the simulation example illustrated in Figure 4-7, an array of performance metrics is available at the conclusion of the output information, encompassing Kappa, FoM (Figure of Merit), and OA (Operational Accuracy). The concluding line provides insight into the temporal duration of the model's execution.It is imperative to note that a warning is present in this example, arising from incongruities in row and column numbers between the restricted file and other dataset files. The model issues a warning when the difference in row and column numbers is within 5; surpassing this threshold results in an error, prompting the model to cease execution. Consequently, meticulous attention to spatial consistency in simulated data during the data preparation phase is indispensable to preempt such warnings and errors.

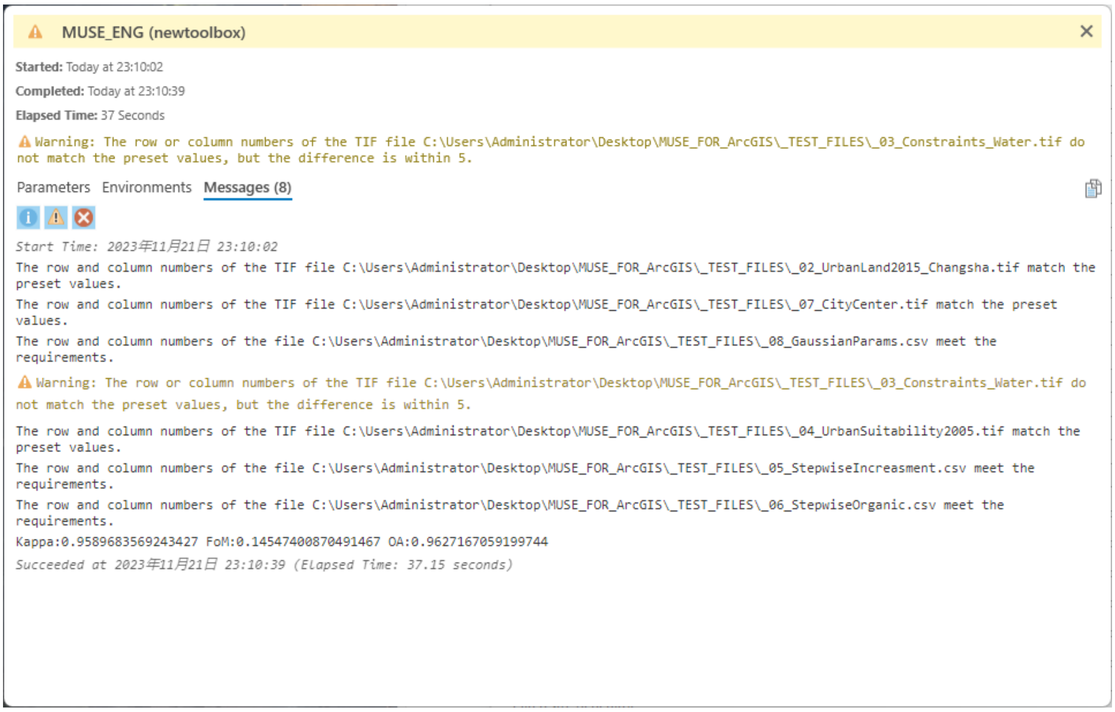

Figure 4-7 Simulation output

4.2.2 Scenario simulation

The procedural steps for parameter input in both scenario prediction mode and model validation mode exhibit similarities but deviate in pivotal aspects. Scenario prediction mode is specifically designed for simulating future urban construction land distributions. Consequently, in this mode, there is no requirement to input urban land data for the simulation end period. Furthermore, when scrutinizing information post-simulation in scenario prediction mode, the assessment of the model's prediction accuracy is notably absent. Refer to Figure 4-8 for a depiction of the interface dedicated to scenario prediction mode.

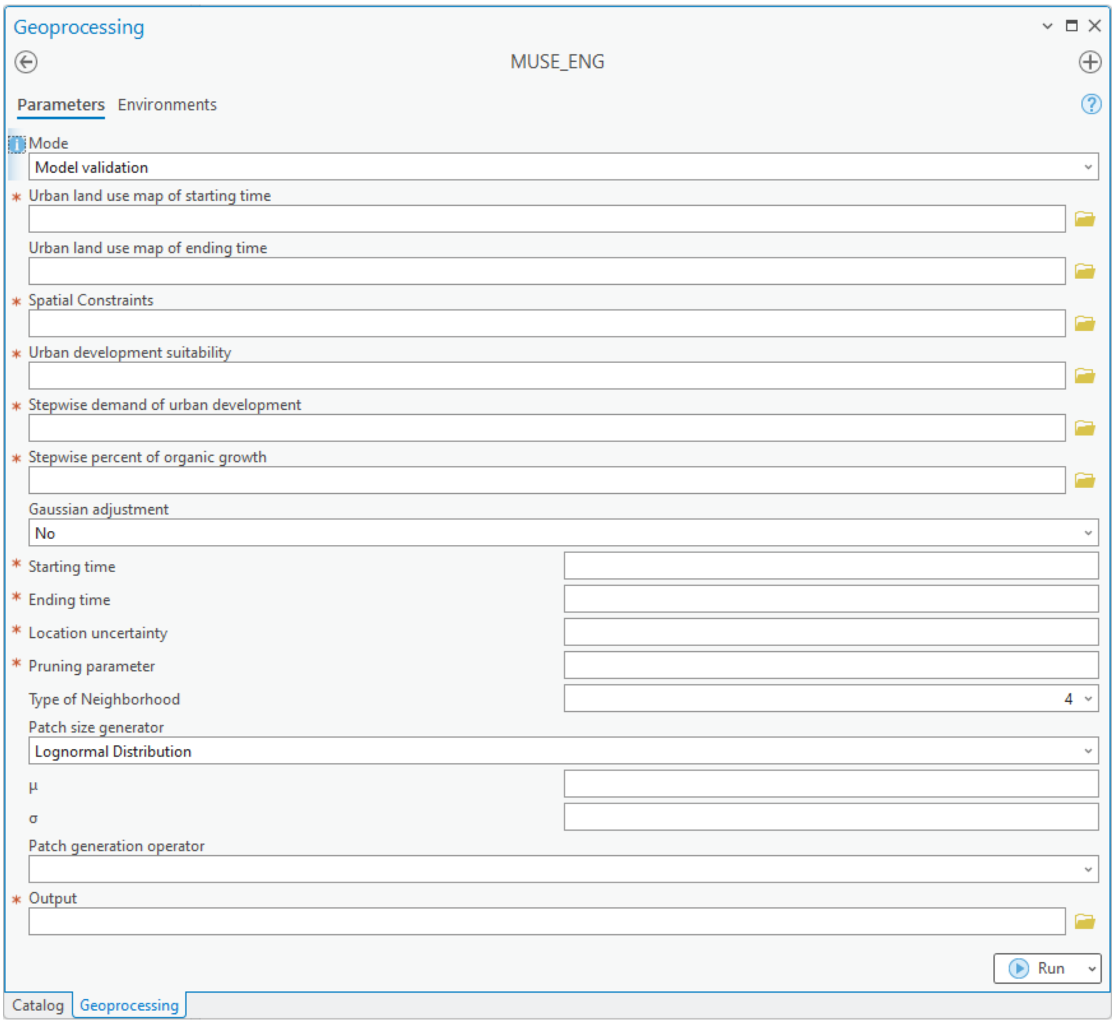

Figure 4-8 Scenario Simulation Interface

4.3 Explanation of simulation results

Following the completion of the simulation, an examination of the results reveals pixel values spanning from 0 to n. Within the symbol system of ArcGIS software, employing a unique value display method, these pixel values manifest sequentially as 0, 1, i…n, where i represents the value designated at the initiation of the simulation incremented by 1, and n corresponds to the value defined at the simulation's conclusion. This configuration provides a lucid depiction of the spatial arrangement of newly allocated urban construction land at each temporal interval. Figure 4-9 serves as an illustrative representation of the simulation outcomes for Changsha spanning from 2005 to 2015.

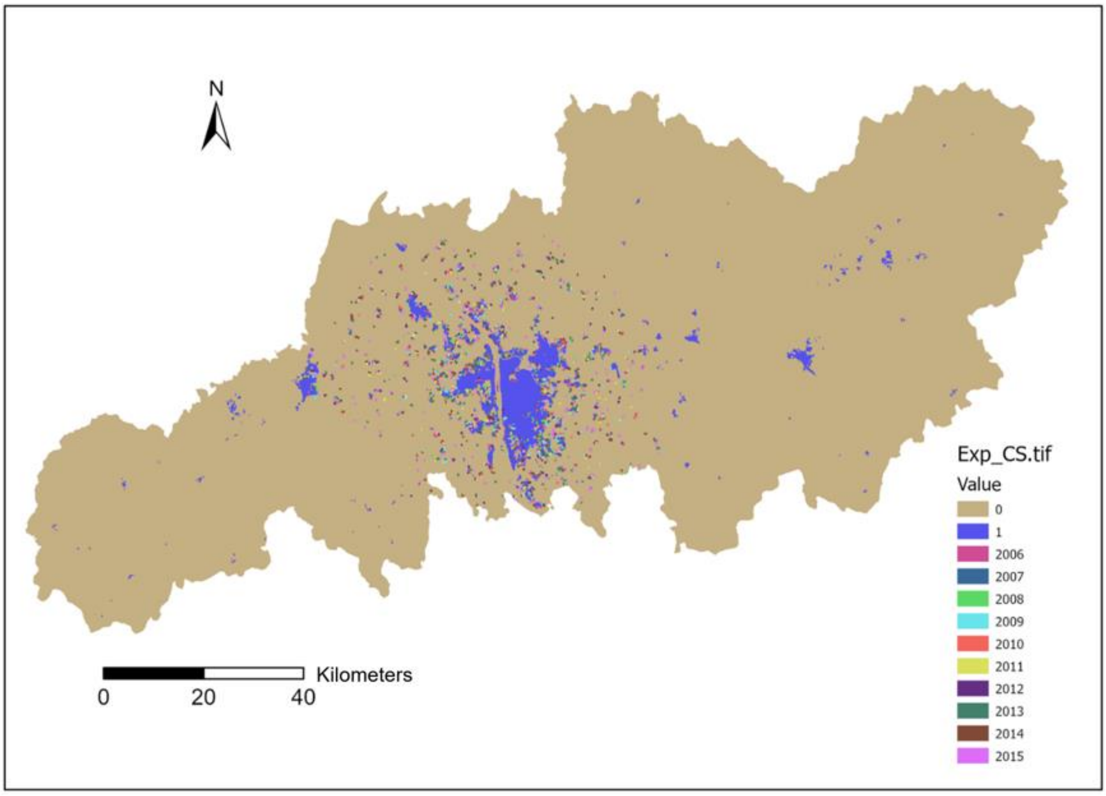

Figure 4-9 Simulation results of urban construction land in Changsha from 2005 to 2015

5. The principles of model

5.1 Basic model framework

The MUSE model adheres to the widely used "quantitative structure-space allocation" simulation framework, and its fundamental operational flow is as follows:

(1) The model reads the total amount of new urban land for each time step.

(2) In each step, MUSE generates various types, sizes, and shapes of urban construction land patches until the total area of generated patches reaches the required amount of new urban land in that time step. To generate these patches, MUSE first determines their type by using a random selection process between organic growth and spontaneous growth patches. The probability of selecting either patch type is controlled by the parameter "organic growth patch ratio", where spontaneous growth patch ratio = 1- organic growth patch ratio. Once determined, MUSE uses user-set controllers to determine the size of each generated patch according to different working principles described in section 5.3.

(3) After determining the size, MUSE generates urban construction land patches with specified areas within the study area using user-set generation engines explained in section 5.4.

(4) These steps are repeated until both types' total area equals that required for that time step's new urban land development.

(5) Then, it simulates subsequent time steps following steps (1)-(4), completing all simulations before saving results.

5.2 Controlling the extent of urban expansion

The urban sprawl degree control module in MUSE is an optional component that regulates the extent of urban sprawl by incorporating a Gaussian function. At its core, this control mechanism calculates the distance between the seed of an urban patch and the city center, utilizing this distance as input for the Gaussian function to derive a correction value. Subsequently, this correction value is weighted and combined with grid suitability to effectively manage the distribution of distances between patches and the city center.

The methods for modifying patch seed suitability are as follows:

$$G(i, d_i, t) = a_t \cdot e^{-\frac{(d-b_t)^2}{\mathrm{c_t}^2}} \quad (1)$$

$$S' = fre \cdot w_0 + S \cdot w_1 \quad (2)$$

$$w_0 + w_1 = 1 \quad (3)$$

The correction value G is determined by the peak value of the gaussian function at time step t, denoted as a_t, which represents the maximum value of the curve. The parameter a_t is always set to 1. The distance from the patch seed to the nearest urban center is represented by d. The mean value of the gaussian function at time step t, denoted as b_t, determines the center position of the curve. The standard deviation of the Gaussian function at time step t, represented by c_t, determines the width of the curve. S and S' represent patch seed suitability before and after correction respectively. w_0 and w_1 represent respectively: attractiveness weight of urban center and suitability weight of urban development.

5.3 Patch size controller

5.3.1 Log-normal distribution patch size controller

The patch generation engine assumes that the area of urban land patches follows a lognormal distribution. MUSE utilizes a random number generator with a lognormal distribution to generate an expected value for the patch size (measured in cells). The implementation of the lognormal distribution area controller is based on the following formula.

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}} \quad (4)$$

Where x is the random variable and represents the obtained value, μ is the mean of the lognormal distribution, and σ is the standard deviation of the lognormal distribution. Specifically, the generated random number x will have the characteristics of a lognormal distribution whose shape is determined by the mean μ and the standard deviation σ.

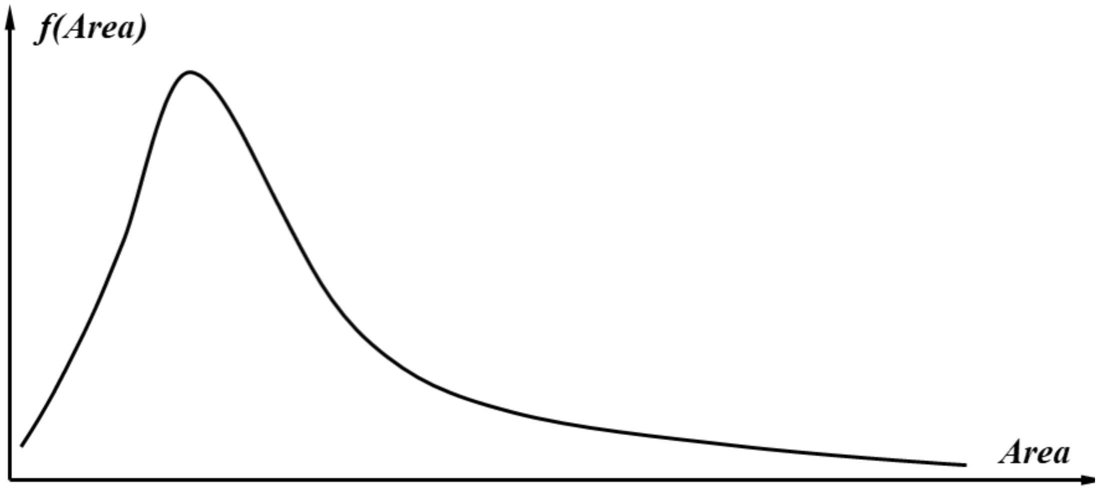

Figure 5-1: Log-normal distribution diagram of patch area

5.3.2 Power law distribution size controller

The power law distribution size controller assumes that the patch sizes follow a power law distribution, with the probability density function given by:

$$S = \alpha_0 (r_\alpha)^{\alpha_1} \tag{5}$$

In the formula, S represents the urban patch area, α_0 and α_1 respectively represent the proportionality constant and power of the power law distribution, and r_α is a random number with a value range between 0 and 1, representing the random probability. Under the power-law distribution, most of the patch area values will cluster in a small range, but there will be a few maximum values, showing a long-tail distribution.

5.3.3 Historical period patch size controller

In this patch size controller, MUSE assumes that the future patch size distribution of urban land could replicate the historical patterns of patch sizes. In other words, it postulates that the distribution of patch sizes for urban development in the future might resemble the distribution patterns observed in historical periods. To utilize this controller, users are required to access historical patch size distribution data for the study area and input this historical patch size data into the controller.

5.4 Patch generation engine

5.4.1 SPGE

The Simple patch generation engine (SPGE) generates patches solely based on the urban development suitability of grid units in the study area. Its fundamental concept is to maximize the urban development suitability of generated patches, while lacking control over their shape. The generation engine operates as follows:

(1) All developable grid units in the study area are sorted from high to low according to their suitability for urban development. Based on the user-defined “pruning factor of patch seed cell database”, grid units are selected from the database to form a patch seed cell database, with these units serving as candidates for initial patch grids.

(2) With the expansion degree switch activated, perform the following operations: Calculate distances between all grids in the patch seed cell library and nearest urban center points. These distances are then substituted into a Gaussian correction function with user-defined

parameters, and results obtained are weighted and summed with original grid suitability to derive modified suitability values. Finally, update suitabilities of grids within the patch seed cell library.

(3) Select an initial grid for each patch from aforementioned patch seed unit library using this method: Randomly choose a grid unit from said library. If it is an organic growth type of patch, ensure spatial connectivity with developed urban grid units; otherwise, re-select and evaluate accordingly. For spontaneous growth type patches, ensure no spatial connectivity exists between chosen unit and developed urban grid units; otherwise repeat selection process. If requirements cannot be met after repeated attempts (3000 times), gradually expand and adjust pruning coefficient of the patch seed unit library.

(4) The raster cells within the initial neighborhood range that are suitable for development are placed into a neighborhood cell library. The neighborhood type can be set by the user as either a 4-neighborhood or 8-neighborhood type. A raster cell is selected from the neighborhood cell library to form a part of the intended generated patch. There are two selection methods:

Random Selection: A raster cell is randomly drawn from the neighborhood library. If the suitability for urban development of the chosen raster cell, which has not been updated, is greater than a random number uniformly distributed between 0 and 1, then the cell is chosen. Otherwise, a new selection is made and assessed.

Maximizing Patch Urban Development Suitability: The raster cell from the neighborhood library with the highest urban development suitability is directly chosen.

The choice between these raster cell selection methods is determined through a dice rolling approach, where the probability of using each method is governed by the parameter “patch position uncertainty”. Specifically, when this parameter is set to 1, only the method of maximizing patch urban development suitability is employed; when set to 0, only the random selection method is used. It’s important to note that if the patch belongs to the self-generated growth type, the selected raster cell from the neighborhood library must not be spatially connected to previously developed urban raster cells. If it is connected, a new selection and assessment are performed.

(5) The selected eligible raster cells from the previous step become a part of the intended generated patch, while the raster cells suitable for development within their neighborhood range are placed into the neighborhood cell library. Another raster cell is chosen from the neighborhood cell library using the methods described above and is added to the growing patch. The suitable raster cells within its neighborhood range are also added to the neighborhood cell library.

(6) Steps (4) and (5) are repeated until the patch reaches the specified size.

(7) The patch seed cell library is restored to the state described in step (1), and then steps (2) to (6) are repeated to generate the next patch. It should be noted that at this point, information about whether raster cells in the patch seed cell library have been developed or whether they are connected to already developed raster cells may change due to the influence of the newly generated patches.

Furthermore, MUSE only performs the operation described in step (1) when starting a new simulation time step. In other words, all urban development patches share the same patch seed cell library within each simulation time step.

Figure 5-2 illustrates the patch growth process of SPGE. For better understanding, we have set the patch position uncertainty parameter to 1, which means only the method of maximizing patch urban development suitability is utilized.

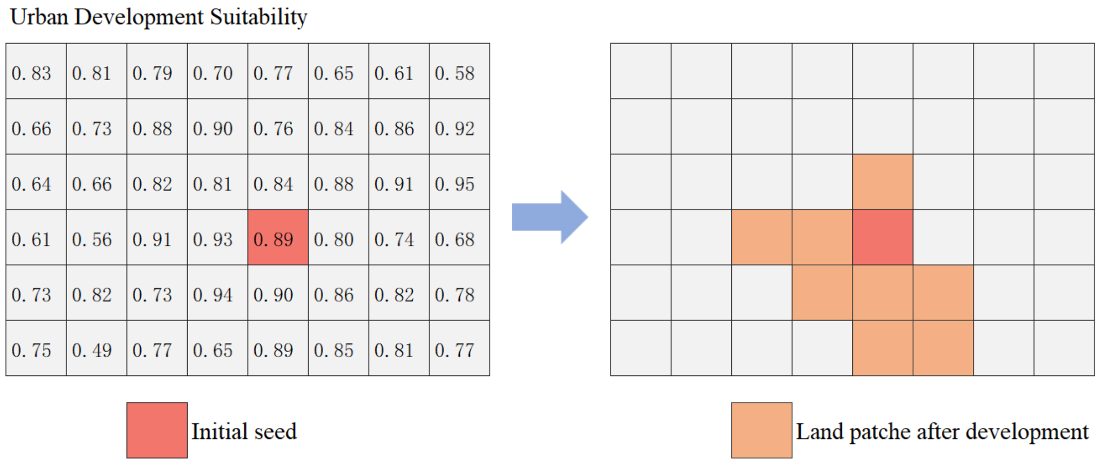

Figure 5-2: Schematic Illustration of SPGE Patch Growth

5.4.2 PPGE

The Parametric Patch Generation Engine (PPGE) builds upon the Simple Patch Generation Engine (SPGE) by incorporating additional patch shape control algorithms. This enhancement allows for control over patch shapes in terms of compactness, continuity, and more. Moreover, it enables a trade-off between maximizing patch urban development suitability and enhancing patch shape by setting different weights. The procedural steps of this engine closely resemble those of the Simple Patch Generation Engine, with the main difference being that the PPGE adjusts the original urban development suitability of each raster cell in the neighborhood cell library based on a patch shape control algorithm. This adjusted suitability is obtained from the input data. Subsequently, the PPGE selects and evaluates raster cells from the neighborhood cell library for inclusion in the intended generated patch based on the modified development suitability.

The method of adjusting urban development suitability is as follows:

$$P_{adjust} = P_{input} \times \mathrm{suit}_{\mathrm{weight}} + \mathrm{Shape}_{\mathrm{score}} \times \mathrm{shape}_{\mathrm{weight}} \quad (6)$$

$$shape_{weight} + suit_{weight} = 1 \quad (7)$$

In the equation:P_{adjust} represents the adjusted development suitability of the raster cell, P_{input} is the development suitability directly read from the input file, $shape_{weight}$ and $suit_{weight}$ are the weight parameters corresponding to the shape control score and the development suitability, respectively.

The implementation of the patch shape control algorithm is based on the following formula:

$$\text{Shape}_{\text{score}} = \frac{\text{diag} - \left(\frac{N + (D - N) \cdot \cos\left(\frac{\theta \cdot A}{2} + O\right)}{D}\right) \cdot \text{dist}}{\text{diag}} \tag{8}$$

In this equation, the patch's shape control score ($\text{Shape}_{\text{score}}$) is determined by considering the distance (dist) and direction (θ, relative to the vertical direction) of each raster cell within the neighborhood to the initial seed cell. These distances and directions can be adjusted using four shape control parameters to obtain the final score. Among these parameters, the numerator (N) and denominator (D) control the ratio between the closest and farthest distances from the seed point to the boundary of the area. Parameter A defines the exponent of the patch, while parameter O determines the direction of the longest axis. The length of the bounding rectangle's diagonal (diag) is used to normalize the shape control score.

Additionally, the patch generation engine introduces two parameters: shape_weight and suit_weight, which are used to adjust the weights of the raster shape control score and urban development suitability. When the shape control score surpasses urban development suitability, the focus of patch growth leans towards shape control. Conversely, when the shape suitability is lower than the development suitability, the emphasis shifts towards maximizing development suitability.

Figure 5-3 illustrates various patch shapes under different parameter settings. It's important to note that during the patch growth process, in order to emphasize shape variations, we have set the patch position uncertainty parameter to 1, indicating the utilization of the method that maximizes patch urban development suitability. For a more comprehensive understanding of the engine's operation principles, please refer to the following publication: https://doi.org/10.1080/136588197242329

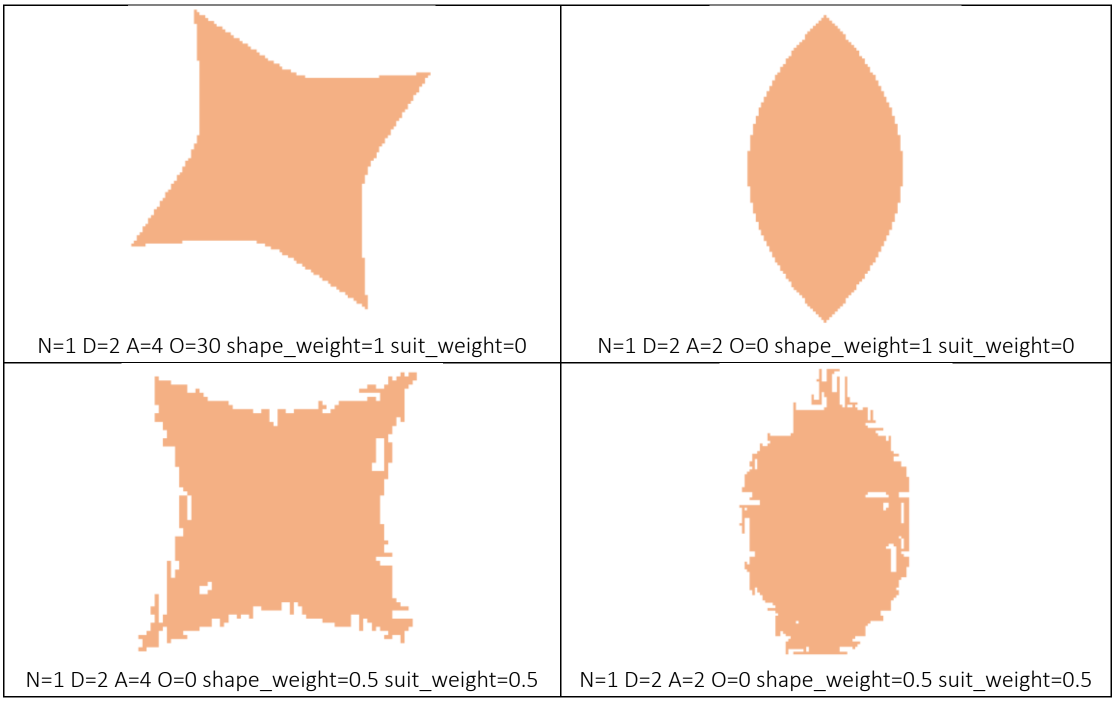

Figure 5-3 Schematic of PPGE Patch Growth

5.4.3 Nei-PGE

The Neighborhood-Controlled Patch Generation Engine (Nei-PGE) is built upon the concept of generating patches based on the urban development suitability of raster cells. It also incorporates adjustments to the development suitability of raster cells based on whether neighboring raster units have already been included in the patch generation process. This adjustment enables control over patch shapes. Specifically, after identifying the initial seed cell, suitable raster cells within the neighborhood range of the seed cell are collected into a neighborhood cell library. This library serves as a pool of candidate cells for the patch to expand from the seed cell.If a newly added neighborhood cell is already present in the neighborhood cell library, the urban development suitability probability of these duplicate neighborhood cells is multiplied by a decimal value beta within the range [0, +∞). When beta belongs to [0, 1], the patch shape tends to diverge from compactness. Conversely, when beta belongs to [1, +∞), the patch shape tends to become more compact and circular. The adjustment of urban development suitability for duplicated neighborhood cells can only occur once. The suitability of cells that are repeatedly added multiple times cannot be adjusted again. The patch generation engine then selects raster cells from the neighborhood cell library based on the adjusted suitability, with the remaining steps mirroring those of the Simple Patch Generation Engine. For a comprehensive understanding of the workings of this engine, please refer to the following publication: https://doi.org/10.1016/j.landurbplan.2022.104640

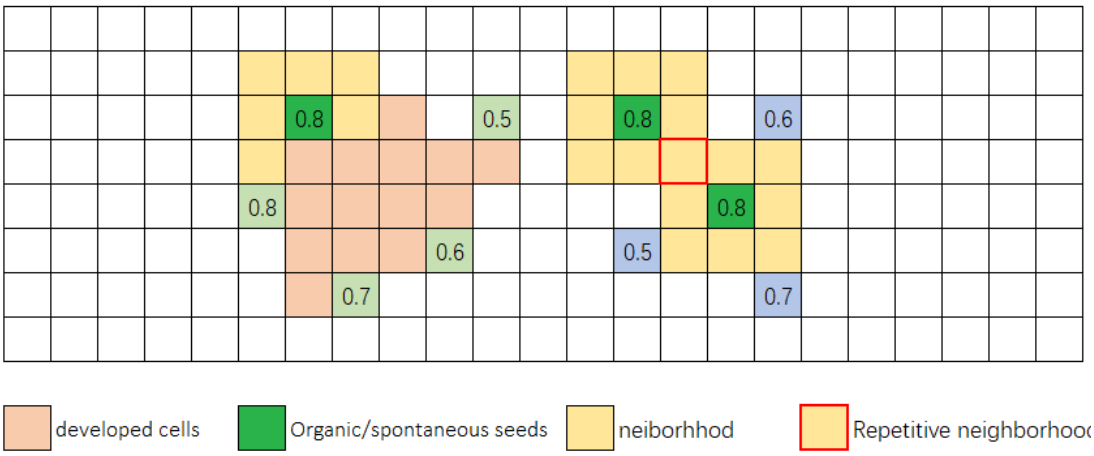

Figure 5-4 Schematic of organic/spontaneous seeds and their neighborhood

5.4.4 Dis-PGE

The Distance-Controlled Patch Generation Engine (Dis-PGE) operates by generating patches based on the urban development suitability of raster cells. Simultaneously, it leverages the Euclidean distance between raster cells in the neighborhood cell library and the initial seed raster cell of the patch to adjust the development suitability of these cells. This adjustment allows for control over patch shapes. The adjustment process is as follows:

$$p' = p \cdot d^{-\delta} \quad (9)$$

Where, p represents the original urban development suitability of the grid unit; p' represents the adjusted development suitability; d is the Euclidean distance between the current cell and the initial seed unit; s is the scale parameter. Specifically, by adjusting the value of s, the engine can influence the shape of urban patches. Increasing s encourages plaque to grow inward, thus promoting the compactness of urban patches. Reducing s, on the other hand, conferred higher development potential on cells further away from the initial seed, resulting in a more dispersed, elongated shape of the urban patch. The patch generation engine selects the grid cells from the neighborhood cell library according to the adjusted suitability, and the other steps are the same as those of the simple patch generation engine. The working principle of the engine can be found in the literature: http://dx.doi.org/10.2139/ssrn.4171720

5.5 Explanation of Important Model Parameters

5.5.1 Proportion of organic growth patches

Adjusting the proportion of organic growth patches can influence the compactness and continuity of urban land layout. When the proportion of organic growth patches is higher, urban expansion tends to connect with existing patches, forming more continuous urban areas. Conversely, when the proportion of organic growth patches is lower, urban expansion leans more towards a self-generated growth pattern. New patches do not connect with the surrounding developed areas, resulting in isolated development zones. Urban development exhibits a more pronounced sprawling expansion state, leading to a more dispersed overall form. Figure 5-5 illustrates the differences between these two types.

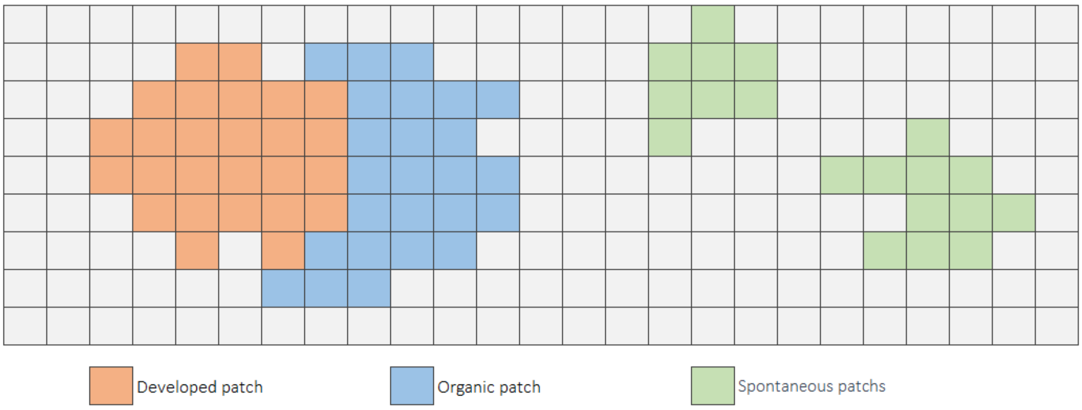

Figure 5-5 Schematic diagram of organic growth patches and spontaneous growth patches

5.5.2 Uncertainty of patch location

Within the patch generation engine, there are two ways to select raster cells from the neighborhood cell library:

Deterministic Selection based on Probability Maximization: This approach involves choosing the cell with the highest suitability (or adjusted suitability) from the candidate neighborhood area as the next growth location.

Uncertain Selection based on Randomization: In this method, a cell is selected by comparing its suitability (or adjusted suitability) with the suitability of a randomly chosen cell from the standardized candidate area. The decision is made probabilistically, akin to rolling dice.

Figure 5-6 illustrates the differences between these two modes.

In the deterministic selection mode, the model tends to pick cells with the highest probability values in the candidate neighborhood area. While this can enhance stability and result repeatability to some extent, it might lead the model to get trapped in local optima, rendering it sensitive to initial conditions and potentially overlooking other potential growth areas. In the stochastic selection mode, randomness is introduced, making the selection process more flexible and diverse. By incorporating randomness, the model can explore a wider range of possibilities in the candidate area, breaking free from the constraints of prior probabilities. This stochasticity can partly mitigate the problem of local optima in deterministic selection, enhancing the model's robustness to initial conditions and making it more capable of global optimization. However, due to the introduction of randomness, the same model parameters might yield different outcomes in different runs, causing irreproducibility in urban growth patterns.

To address the challenges posed by these two modes, we introduced a patch position uncertainty parameter within the patch generation engine. This parameter influences the balance between deterministic and stochastic selections in the urban growth pattern. By adjusting this parameter, one can flexibly control the proportion of deterministic and stochastic selections, thereby influencing the overall behavior of the model. When the parameter is set to a higher value, the model is inclined towards the deterministic selection mode, enhancing stability and predictability. Conversely, a lower parameter value will give dominance to the stochastic selection mode, allowing the model to engage in more random exploration during patch selection.

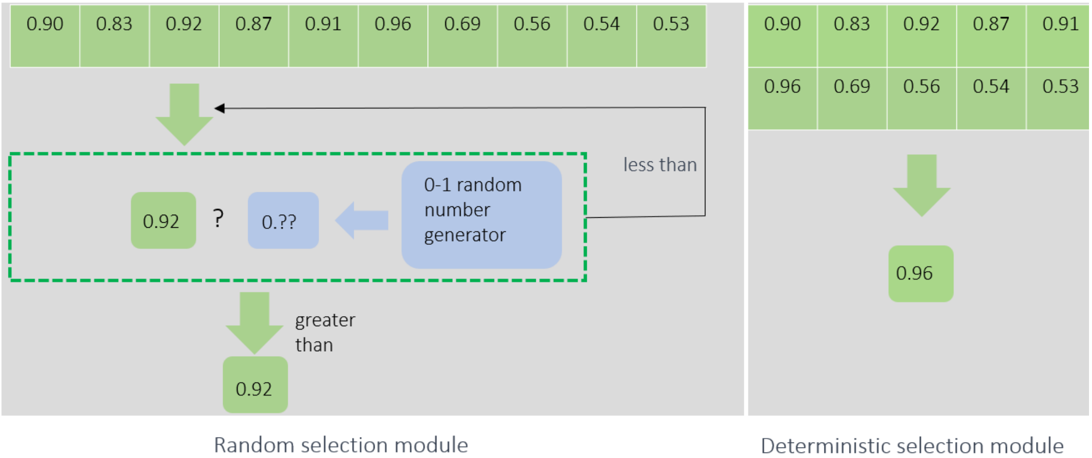

Figure 5-6 Schematic diagram of random selection and deterministic selection

5.5.3 Pruning coefficient for patch seed library

In the process of constructing the initial seed cell library for this model, MUSE arranges all undeveloped raster cells within the study area in descending order based on their individual urban development suitability. This forms an untrimmed seed cell library. Subsequently, MUSE introduces a user-defined decimal value between 0 and 1, referred to as the seed cell library pruning coefficient. This coefficient is employed to control the size of the initial seed cell library to be considered during the seed cell selection process. Specifically, we use the following formula to calculate the size of the pruned seed cell library:

$$A = Len \cdot T \tag{10}$$

In this context, A signifies the size of the pruned seed unit library, delineating the scope within which seed units are selected. Len represents the total count of all exploitable grid units in the study area, organized in descending order of suitability. This count reflects the overall pool of exploitable grid units available. T corresponds to the seed unit library pruning coefficient—a parameter influencing the dimensions of potential urban development zones. By fine-tuning the value of T MUSE gains the versatility to manage the extent of selected exploitable areas.

For lower T values, the initially chosen seed units gravitate toward areas with higher suitability, directing urban development primarily towards regions with superior suitability scores. Consequently, zones with comparatively lower suitability are excluded from the selection of exploitable regions. This configuration yields a more compact simulated spatial distribution of urban construction land. Conversely, as T approaches 1, even seed units with lower development probabilities stand a chance of selection. This expansion results from the potential urban land patches encompassing the entire exploitable area of the study region. In this scenario, the simulated spatial distribution of urban construction land tends to adopt a more dispersed pattern.

6. Copyright statement and contact information

Multi-engine Urban Expansion Simulator Toolbox Vesion 1.0

Designers: Jianxin Yang, Rui Shi, Dr. Wenwu Tang

Developers: Rui Shi, Jingruo Wang, Jianxin Yang

Contact infomation：

- Yang Jianxin- School of Public Administration (yangjianxin@cug.edu.cn)
- shirui@cug.edu.cn
- qaz1200a@163.com
- Dr. Wenwu Tang | Home (charlotte.edu) (WenwuTang@uncc.edu)