# Multi-engine Urban Expansion Simulator （MUSE） User's Manual

2023/07

China University of Geosciences (Wuhan)

https://www.landinsightlab.com/

# contents

# 1.Introduction

The MUSE (Multi-engine Urban Expansion Simulator) is a cellular automata-based model designed for simulating urban growth. It incorporates three patch size generators and four different patch generation techniques. Its main goal is to faithfully imitate the patterns and procedures involved in urban land expansion. The following are the four patch generating algorithms:

Distance-Constrained Patch Shape (Dis-PGE);

Neighborhood-Constrained Patch Shape (Nei-PGE);

Neighborhood-Constrained Patch Shape (SPGE);

Parameterized Patch-Growing Engine with Tradeoff between Maximized Cell Suitability and Optimized Patch Shape (PPGE).

n addition, MUSE provides three patch size controllers for patch size determination:

Lognormal Distribution: This Patch Size Controller operates under the assumption that patch areas follow a log-normal distribution (refer to Equation (4)).

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

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

With MUSE, users can create multiple spatial patterns for urban land structure, construct patches of varying sizes and forms using different algorithms and generators, and mimic the process of urban expansion. This model has great potential for use in decision support, land resource management, urban planning, and land use planning..

# 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 | 4GB of RAM or more |
| Storage Space | At least 10GB of available space |

# 3.Data Preparation

To access the installation directory of MUSE, navigate to a location such as C:\Program Files\MUSE_.Test_Files. Within this folder, you'll find a total of 9 files comprising 4 CSV files and 5 TIFF files. These files are examples and are necessary for the operation of MUSE. The data preparation techniques utilizing the aforementioned sample files will then be covered in the documentation.

- _TEST_FILES
  - _01_UrbanLand2005_Changsha.tif
  - _02_Constraints_Water.tif
  - _05_UrbanSuitability2005.tif
  - _06_UrbanLand2015_Changsha.tif
  - _07_CityCenter.tif
  - _03_StepwiseOrganic.csv
  - _04_StepwiseIncreasment.csv
  - _08_GaussianParams.csv
  - _09_History_CellSize.csv

Figure 3-1: Sample File Directory

## 3.1 CSV documents

### *3.1.1 Stepwise demand of urban development*

The stepwise urban development demand incremental data is stored in a CSV file with two columns. The first column displays several simulated time points. The time unit in this case is years, but it could be any other time unit. The second column shows the proportionate increase in the number of urban land grid cells at each time point. These grid cell values, presented in grid cells, demonstrate how much urban land area expanded within the research region over each period (the time interval between consecutive time periods). The example file "_04_StepwiseIncrement.csv" records the yearly rise in the number of urban land grid cells and the proportion of organic patches in the research region between 2005 and 2015.

| | A | B | C |
|---|---|---|---|
| 1 | **year** | **cellNum** | **organic** |
| 2 | urban_area_2006 | 15403 | 0.4 |
| 3 | urban_area_2007 | 18573 | 0.5 |
| 4 | urban_area_2008 | 20348 | 0.63 |
| 5 | urban_area_2009 | 23048 | 0.67 |
| 6 | urban_area_2010 | 25043 | 0.66 |
| 7 | urban_area_2011 | 27583 | 0.71 |
| 8 | urban_area_2012 | 29746 | 0.75 |
| 9 | urban_area_2013 | 39325 | 0.72 |
| 10 | urban_area_2014 | 43048 | 0.72 |
| 11 | urban_area_2015 | 57355 | 0.77 |

Figure 3-2: Stepwise demand file for urban development

### *3.1.2 History patch size controller data*

The patch size controller data is saved in CSV format, with two columns in each row. The first column represents the patch number for a certain historical time, while the second column represents the patch area size. In the file, these patch sizes are stored as integers, with each row showing the number of grid units covered by the patch. As shown in Figure 3-3, the sample file "_09_History_CellSize.csv" records 5000 patch sizes.

| | 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 Parcel Sizes in Historical Periods

### *3.1.3 Gaussian correction parameter data*

The Gaussian correction parameter data is stored in CSV format and consists of three columns. The first column represents different simulation time points. The second and third columns contain data corresponding to the mean parameter 'b' (measured in km) and the standard deviation parameter 'c' (also in units of km) of the Gaussian function, respectively. These values may be in decimal or integer format. The sample file "_08_GaussianParams.csv" documents the values of parameters 'b' and 'c' for the Gaussian correction within the time span from 2005 to 2015, as illustrated 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

### *3.1.4 Stepwise percent of organic growth data*

This data is stored in CSV format, comprising two columns. The first column denotes different simulation time points, while the second column represents the associated organic patch area ratios at each time point. These ratios are stored as decimals ranging between 0 and 1. It's important to note that the area ratio of spontaneously grown patch types is equivalent to 1 minus the area ratio of organic patches. The sample file "_03_StepwiseOrganic.csv" documents the area ratios of newly added organic patches spanning from 2005 to 2015, as depicted in Figure 3-5.

| year | organic |
|---|---|
| urban_area_2006 | 0.7 |
| urban_area_2007 | 0.75 |
| urban_area_2008 | 0.77 |
| urban_area_2009 | 0.75 |
| urban_area_2010 | 0.73 |
| urban_area_2011 | 0.71 |
| urban_area_2012 | 0.72 |
| urban_area_2013 | 0.7 |
| urban_area_2014 | 0.77 |
| urban_area_2015 | 0.79 |

Figure 3-5: Example File of Organic Patch Growth Ratios

## 3.2 TIF files

Users must provide 5 TIF files containing specific data required for MUSE program operation. These files provide spatial information such as the distribution of urban construction land at the base and model validation simulation sites, the probability of urban construction appropriateness, urban development construction constraints, and urban center point data. These files must retain rigorous consistency in spatial features, such as the same number of rows and columns, projection coordinates, and spatial resolution. Figure 3-6 shows an example file. Here is an outline of the function and format of each file:

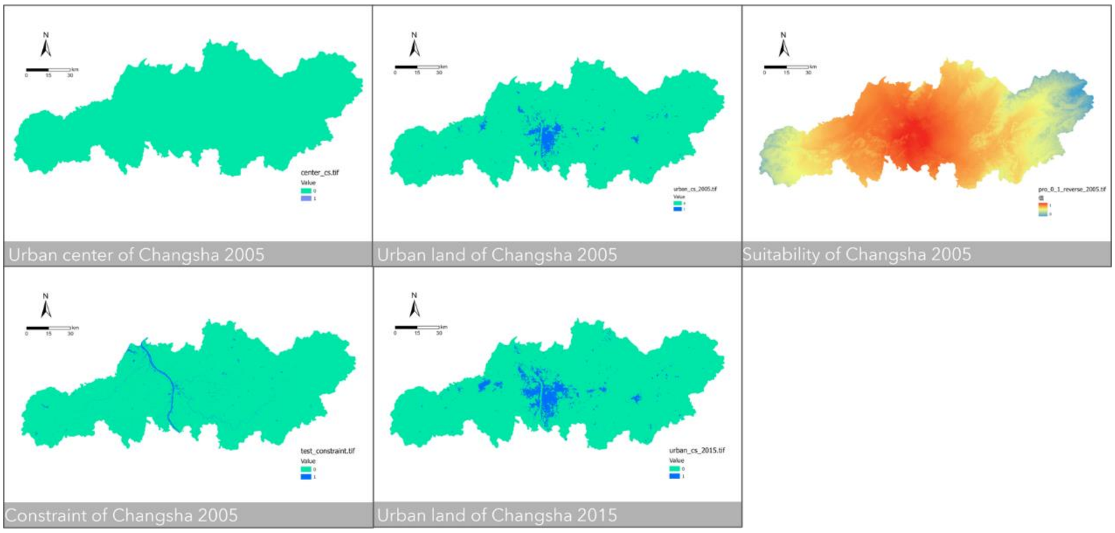



Figure 3-6: model input data files

(1) Base Period Urban Construction Land Data: Describes the spatial distribution of urban construction land during the baseline period. Values are denoted by 0 and 1, where 0 signifies undeveloped land and 1 represents developed urban areas.

(2) Model Validation Simulation Urban Construction Land Data: Reflects the urban construction land distribution at the validation simulation point. Similar to the base period data, values range from 0 to 1, representing undeveloped and developed land. It is used to compare and evaluate simulation accuracy against MUSE output data.

(3) Urban Construction Suitability Probability File: Represents the probability of urban construction suitability based on diverse driving factor data. Values in this file range from 0 to 1, where higher values indicate higher suitability for urban development.

(4) Urban Development Construction Restriction File: Provides information on spatial constraints for urban development. It identifies areas with restrictions based on simulation scenario requirements, such as water bodies, protected farmland, historical and cultural conservation areas, etc. Values are expressed as 0 for open development areas and 1 for restricted development areas.

(5) Urban Center Point Data: Specifies the locations of urban center points, typically situated in areas like central business districts (CBD), government institutions, or key city hubs. Values vary between 0 and 1, with 0 representing non-center grid cells and 1 denoting center grid cells. Each urban center is denoted by one grid cell, ensuring consistency between the number of grid cells with a value of 1 and the count 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_UrbanLand2005_Changsha.tif |
| Model Validation Urban Construction Land Spatial Distribution Data | Remote Sensing Data, Land Survey Data, etc. | Remote Sensing Data, Land Survey Data, etc. | _06_UrbanLand2015_Changsha.tif |
| Urban Construction Suitability Probability File | Evaluated based on Driver Factor Data | 0-1 (Urban Construction Suitability Probability) | _05_UrbanSuitability2005 .tif |
| Urban Development and Construction Restriction File | Set based on Real Conditions and Simulated Scenario Requirements | 0 (Developable Area), 1 (Restricted Development Area) | _02_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

## 4.1 Software Installation

(1) Double-click the software installation package MUSE-1.0-Setup.exe and select the installation mode. It is recommended to choose "Install for all users".

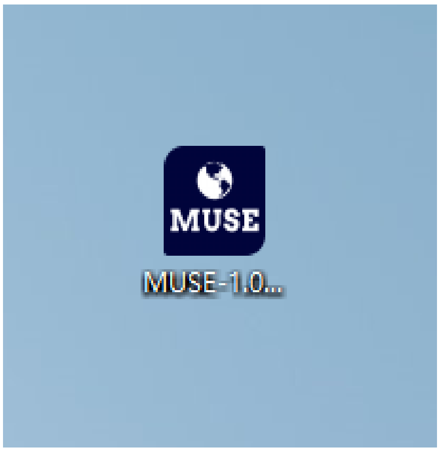



Figure 4-1: MUSE Shortcut

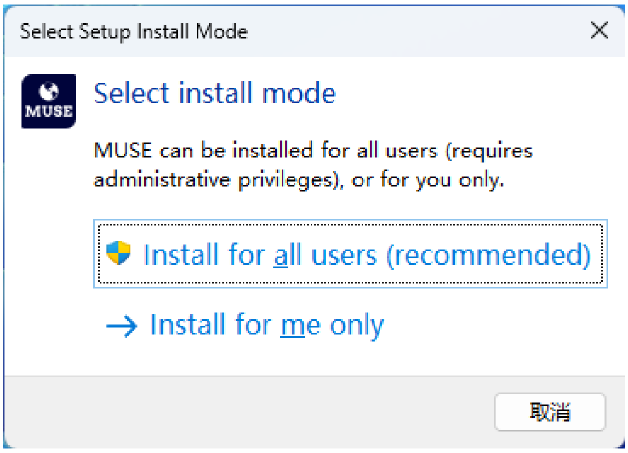



Figure 4-2: Installation Mode Selection

(2) Customize the installation directory (default or modified) and select the Start menu folder.

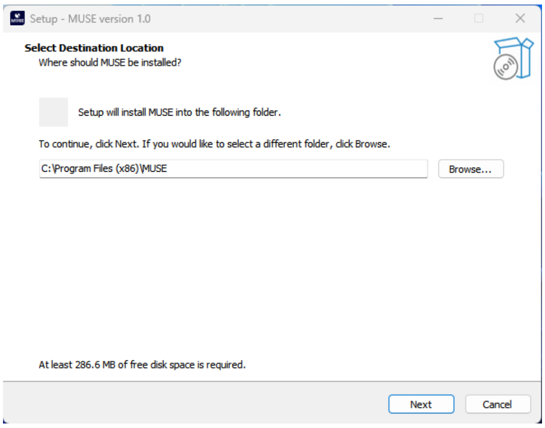



Figure 4-2: Installation Mode Selection

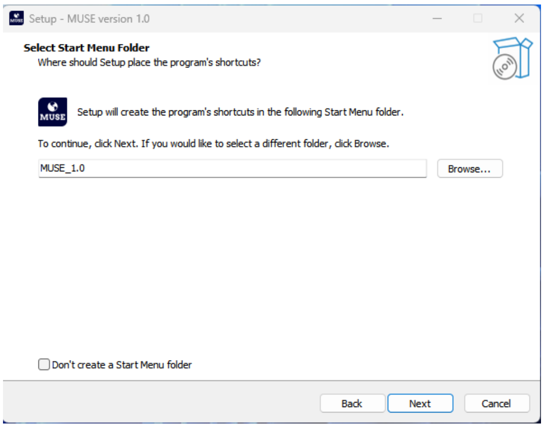



Figure 4-4: Start Menu Folder Selection

(3) Check the box for "Create desktop shortcut," then verify the installation information and click on "Install" to proceed.

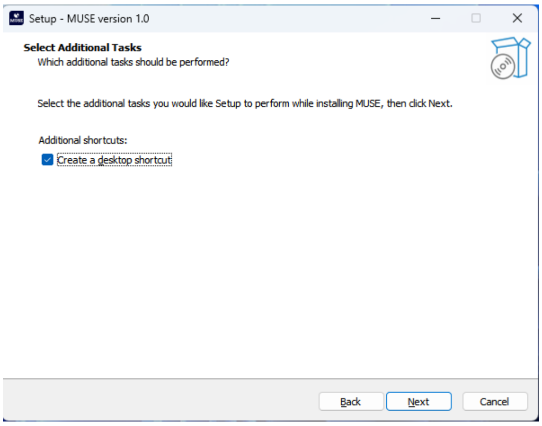



Figure 4-5: Desktop Shortcut Creation Selection

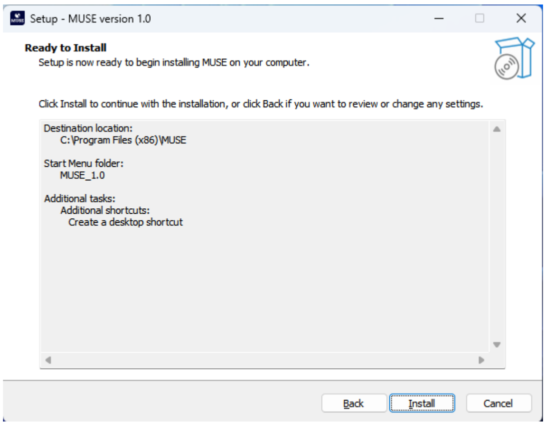



Figure 4-6: Installation Information Confirmation

(4) After the installation progress is complete, click Finish to complete the installation.

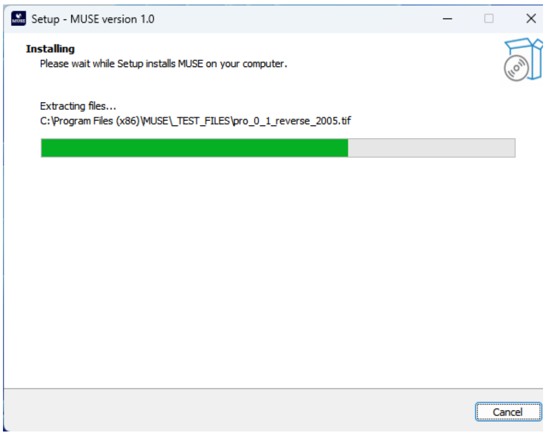



Figure 4-7: Installation Progress

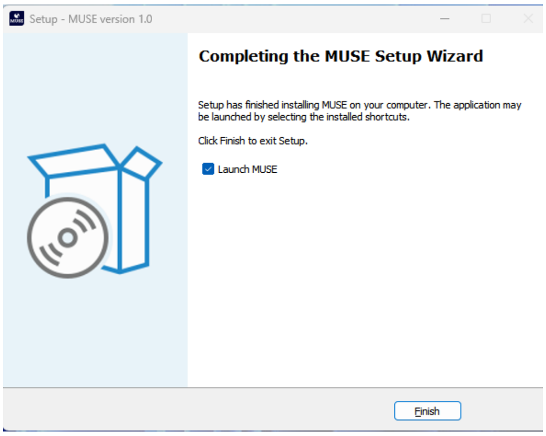



Figure 4-8: Installation Completed

## 4.2 Running the Software

### *4.2.1 Model Validation*

Step 1: Find the MUSE shortcut, double-click to run it, and access the main program interface as shown in Figure 4-9.

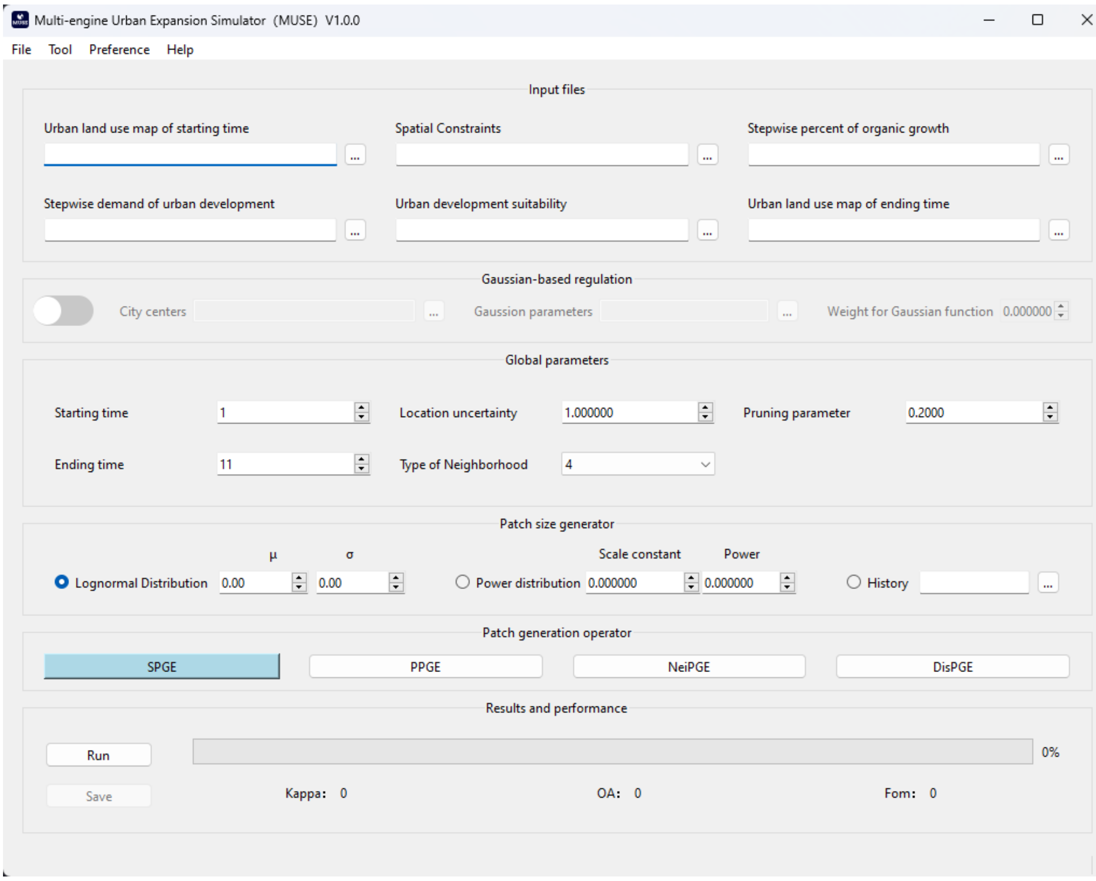



Figure 4-9: The main interface of MUSE.

Step 2: Enter the required data for the model in the input data area, inputting each dataset as per the corresponding relationship outlined in Table 2.

Table 2: Overview of Input File Information

| Parameters Name | File Types | Example Files |
| --- | --- | --- |

| | | |
|---|---|---|
| Simulated Initial Urban Land | Base-year Urban Construction Land Data | _01_UrbanLand2005_Changsha.tif |
| Simulated Final Urban Land | Target-year Urban Construction Land Data | _06_UrbanLand2015_Changsha.tif |
| Development Restriction Factors | Urban Development Restriction File | _02_Constraints_Water.tif |
| Urban Construction Suitability | Urban Construction Suitability Probability File | _05_UrbanSuitability2005.tif |
| New Urban Land Area | Urban Construction Land Increment | _04_StepwiseIncreasment.csv |
| Stepwise percent of organic growth | Patch Organic Growth Category Ratio Data | _03_StepwiseOrganic.csv |

Step 3: Please indicate if you want to use the Urban Expansion Control Module. If you want to use this module, toggle the switch on the left side of the space to enable it. Enter data for urban center points, Gaussian correction parameters, and the weight showing the impact of the urban center on urban land development (value range: 0-1). If you do not want to use this functionality, simply turn it off. Table 3 shows the relationships for each dataset.

Table 3: Overview of Input File 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 |

Step 4: It is critical to configure five vital parameters under the model's global parameter section. Specifically, during setup, the overall duration of the simulation period (simulation finish - simulation start) must comply to certain guidelines: its value must not exceed the timeframe indicated in the urban construction land increment data. To be more specific, this value is calculated by subtracting one from the total number of rows in the urban construction land increment data, which represents the duration. For example, if the urban development land increment data (including headers) contains 11 rows of information spanning the years 2006 to 2015, the aforementioned duration parameter would be 10 years. As a result, when defining the simulation period length, it should be less than or equal to 10 years. Essentially, the time difference between the simulation's finish and beginning should not be more than ten years. The table below details the definitions and acceptable value ranges for each parameter in the global parameter section.

Table 4: Model Parameters

| Parameter Name | Parameter 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 |

Step 5: Please select one of the three patch size generators. Among these options, "lognormal distribution" and "power law distribution" are predefined generation strategies integrated into the model. They assume that the sizes of newly generated patches adhere to either a lognormal or power law distribution, respectively. If you opt for the Historical Period Patch size, you'll be required to furnish a CSV file containing customized patch sizes formatted as described in section 3.1. The "lognormal distribution" strategy employs a random generator based on the lognormal distribution to regulate the sizes of generated patches. This procedure involves entering parameters such as the lognormal distribution's expectation and logarithmic standard deviation. Conversely, the "power law distribution" strategy utilizes a random generator based on the power law distribution to control patch area sizes. To govern the distribution of patch sizes, this approach requires the input of parameters such as the proportional constant and power.

Step 6: Choose the patch generation engine. MUSE provides four algorithms for patch generation engines. When you pick the Neighborhood Controlled Patch Generation Engine, the model will automatically set the patch position uncertainty parameter to 1 and the neighborhood type parameter to 8. This adjustment is due to the Neighborhood Controlled Patch Generation Engine's mechanism, which is based on a stochastic process and a repeating neighborhood mechanism. The following table details the parameters for each engine:

Table 5: Explanation of Engine Control Parameters

| **Explanation of Engine Control Parameters** | | | | |
|---|---|---|---|---|
| **Engine Name** | **Parameter Name** | **Parameter Description** | **Default Value** | **Value Range** |
| SPGE | This engine does not require any input parameters. | | | |
| PPGE | N | N and D together influence the | 1 | Greater than 0 |
| | D | longest dimension of the plaque | 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 |

Step 7: To initiate the model execution, click on the 'Start Simulation' button. The program will transition into the running state, and upon completion of the simulation, the 'Save Simulation Data' button will shift from being inactive to an active clickable state, as illustrated in Figure 4-10.

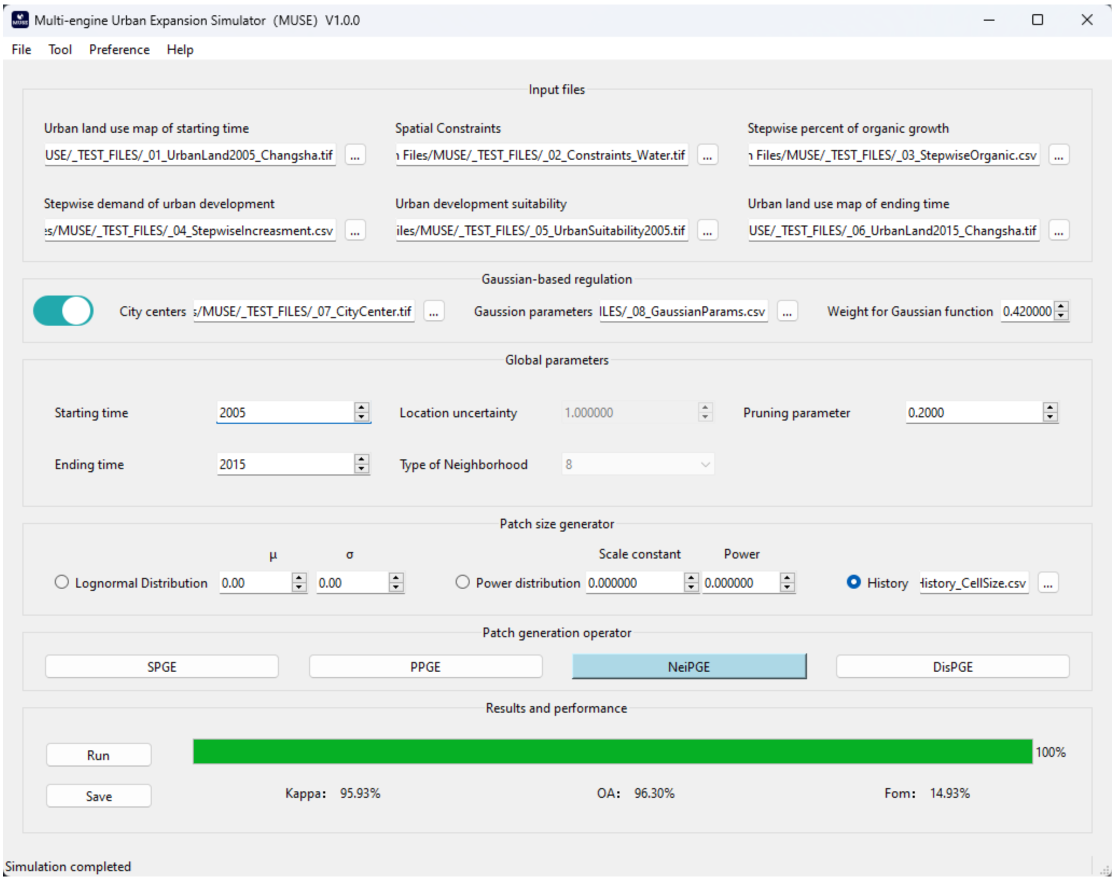



Figure 4-10: The interface after the execution has completed

Step 8: To assess the simulation accuracy, MUSE offers three indices: Kappa, Operation Accuracy (OA), and Figure of Merit (FOM). These indices are utilized to characterize the model's simulation accuracy.

Step 9: If the model's simulation results meet the accuracy requirements, you can select 'Save Simulation Data.' After specifying the storage path and file name for the simulation results, you can save the model's output from the simulation.

### *4.2.2 Scenario simulation*

After completing model verification, users can access the scenario prediction mode by navigating to "Preferences" > "Mode" > "Scenario Prediction". The operational process within this mode mirrors that of the model verification mode. However, in the scenario prediction module, users are not required to input urban land data at the simulation's conclusion, and the model does not display relevant accuracy indicators. Refer to Figure 4-11 for the interface of the scenario prediction module.

When the program enters the scenario prediction mode, it uses the urban land data observed at the end of the simulation verification stage (observation data, not simulation results) as the initial urban land data for the prediction stage. Similarly, the simulation time period within the model parameters automatically moves from past to future time points.

The necessary inputs for users in scenario prediction include data on development limitation factors, new urban land areas, and urban construction suitability. Users should adjust these inputs based on their forecasts and expectations concerning future urban development. Users also have the option to modify other parameters to simulate various scenarios or retain parameters utilized during the model validation phase.

Once these preparations are finalized, users simply need to click on the "Run" button to commence the simulation for scenario prediction. After the simulation process is completed, users can save the predicted results at a specified location by clicking on "Save Simulation Data".

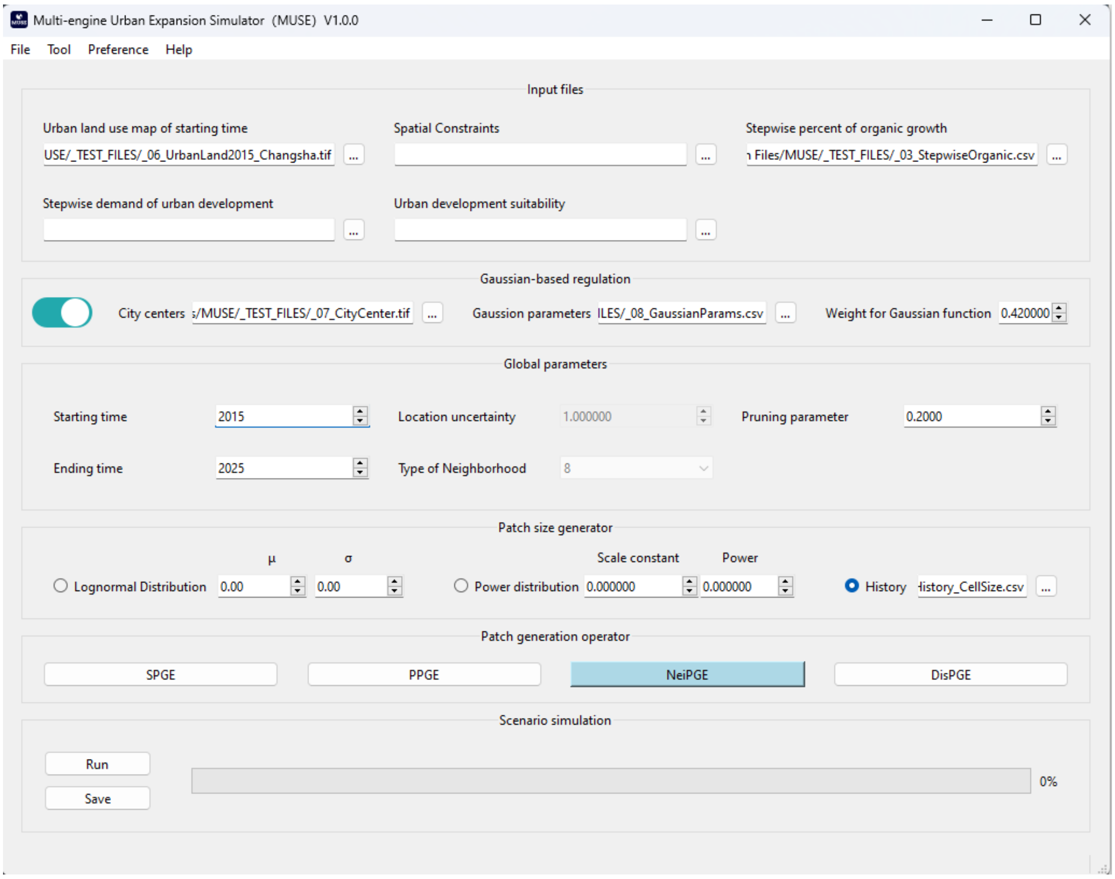


Figure 4-11: Scenario Simulation Interface

## 4.3 Saving and opening parameter files

Users have the option to save the configured input files and various parameters by using the 'File- Save' function, creating a .mud file for convenient access in subsequent operations.

This method eliminates the necessity to reset parameters individually. The saved parameter file can be accessed and loaded via the 'File- Open' function.

### 4.3 Explanation of simulation results

Upon reviewing the completed simulation results, you'll notice that the pixel values range from 0 to 'n'. In the symbol system of ArcGIS software, unique values are displayed sequentially, presenting pixel values as 0, 1, i...n. Here, 'i' represents the initial value set at the beginning of the simulation plus 1, while 'n' represents the value set at the end of the simulation. This setup provides a clear insight into the spatial arrangement of newly added urban construction land at each time step. Figure 4-12 showcases the simulation results from 2005 to 2015 for Changsha City.

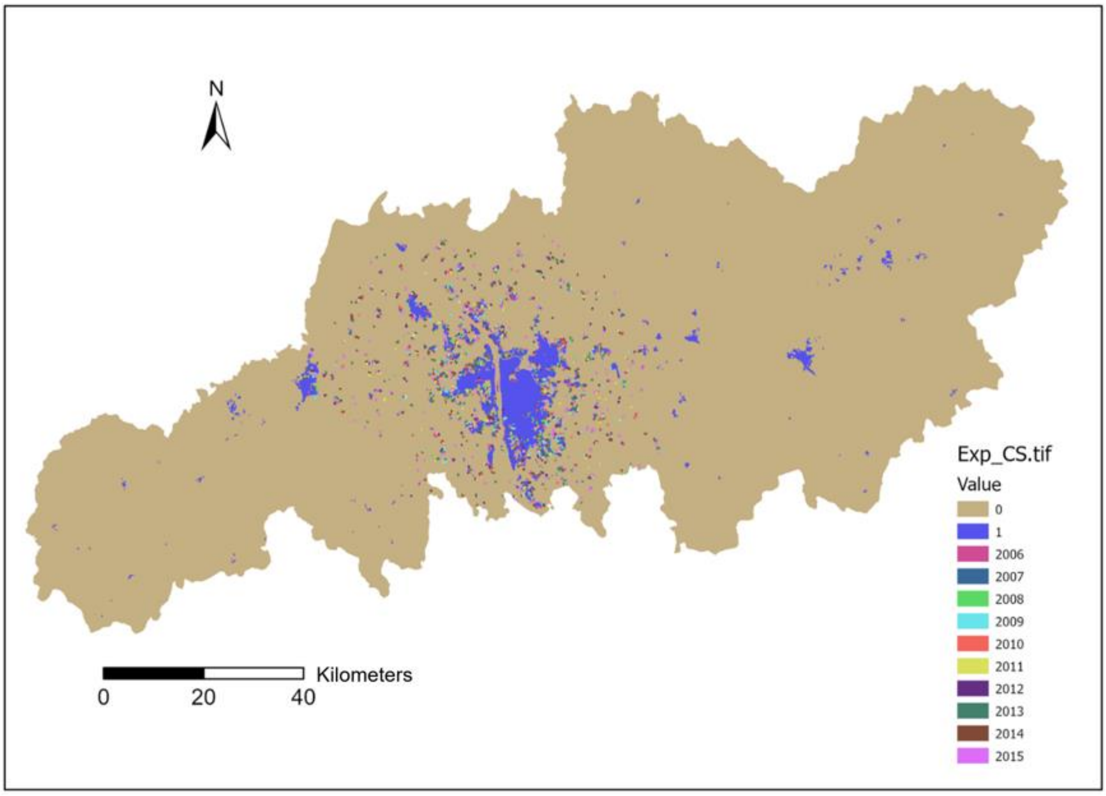



Figure 4-12: Simulation results of urban construction land in Changsha from 2005 to 2015

## 5. The principles of model

### 5.1 Basic model framework

The MUSE model operates within the well-established "quantitative structure-space allocation" simulation framework. Its operational flow can be outlined as follows:

(1) The model retrieves the total volume of new urban land designated for each time step.

(2) At each step, MUSE generates diverse types, sizes, and configurations of urban construction land patches until the collective area of generated patches fulfills the required volume of new urban land for that particular time step. To create these patches, MUSE initially determines their type through a random selection process between organic growth and spontaneous growth patches. The likelihood of selecting a specific patch type is regulated by

the "organic growth patch ratio" parameter, where the spontaneous growth patch ratio equals 1 minus the organic growth patch ratio. Subsequently, MUSE employs user-defined controllers to determine the size of each generated patch, following distinct principles described in section 5.3.

(3) Upon determining the patch sizes, MUSE generates urban construction land patches with designated areas within the study area, employing user-specified generation engines explained in section 5.4.

(4) These procedures iterate until the total area of both patch types matches the requirement for new urban land development in that particular time step.

(5) Subsequently, the model progresses to simulate subsequent time steps by following steps (1)-(4) iteratively, completing all simulations before storing the results.

## 5.2 Controlling the extent of urban expansion

The Urban Sprawl Degree Control module in MUSE is an optional component that integrates a Gaussian function to control the magnitude of urban sprawl. This control system is primarily concerned with calculating the distance between the seed of an urban patch and the city center. This distance is fed into the Gaussian function, which produces a correction value. This adjustment value is then weighted and combined with grid suitability data to influence 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 operates under the assumption that the urban land patch sizes conform to a lognormal distribution. MUSE employs a random number generator that adheres to a lognormal distribution to calculate an anticipated value for the patch size, measured in cells. The lognormal distribution area controller is implemented based on the following formula.

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

Where $x$ is the random variable and represents the obtained value, $\mu$ is the mean of the lognormal distribution, and $\sigma$ 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 $\mu$ and the standard deviation $\sigma$

.

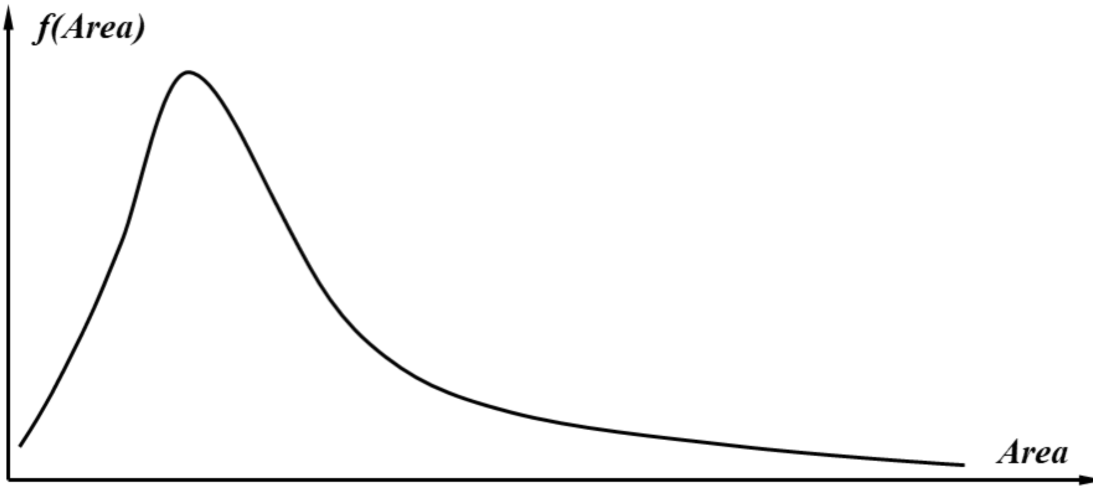


Figure 5-1: Log-normal distribution diagram of patch area

### *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, $\alpha_0$ and $\alpha_1$ respectively represent the proportionality constant and power of the power law distribution, and $r_\alpha$ 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*

MUSE hypothesizes in this patch size controller that the future distribution of urban land patch sizes will follow previous patch size trends. Essentially, it implies that the distribution of patch sizes in future urban development may reflect previous trends. To use this controller properly, users must get historical patch size distribution data specific to the research area and enter it into the controller.

## 5.4 Patch generation engine

### *5.4.1 SPGE*

The Simple Patch Generation Engine (SPGE) generates patches only based on the suitability of grid units for urban development within the study area. It seeks to maximize the suitability of generated patches for urban development, with suitability taking precedence over shape control. The engine works like this:

(1) Selection of Grid Units:

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) Expansion and Suitability Calculation:

Perform the following operations with the expansion degree switch turned on: Calculate distances between all grids in the patch seed cell library and nearest urban center points. These distances are then entered into a Gaussian correction function with user-defined parameters, and the resulting results are weighted and averaged with the original grid suitability to yield adjusted suitability values. Finally, within the patch seed cell library, change grid suitability.

(3) Initial Patch Selection:

Using the following procedure, choose an initial grid for each patch from the patch seed unit library: Choose a grid unit at random from the library. If it is an organic growing patch, make sure it has spatial linkage with developed urban grid units; otherwise, re-select and analyze. For patches with spontaneous development, make sure there is no spatial connectivity between the chosen unit and the created urban grid units; otherwise, repeat the selection procedure. If the requirements are not reached after 3000 trials, progressively extend and alter the pruning coefficient of the patch seed unit library.

(4) Neighborhood Cell Library and Raster Cell Selection:

The raster cells that are suitable for development within the initial neighborhood range are added to a neighborhood cell library. The user can choose between a 4-neighborhood and an 8-neighborhood neighborhood type. To form part of the intended created patch, a raster cell is chosen from the neighborhood cell library. There are two methods of selection:

Random Selection: A raster cell is picked at random from the local library. If the suitability of the chosen raster cell for urban development is greater than a random value randomly distributed between 0 and 1, the cell is picked. Otherwise, a new choice is created and evaluated.

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

The decision between these raster cell selection methods is made using a dice rolling methodology, with the likelihood of adopting each method set by the parameter "patch position uncertainty." When set to 1, only the approach of maximizing patch urban development suitability is utilized; when set to 0, only the random selection method is used. It is vital to remember that if the patch is self-generated, the raster cell chosen from the neighborhood library must not be physically related to previously constructed urban raster cells. If it is connected, a new selection and evaluation are carried out.

(5) Patch Expansion:

The previously selected eligible raster cells are incorporated into the intended created patch, while the raster cells acceptable for development within their neighborhood range are added to the neighborhood cell library. Using the procedures outlined above, another raster cell is selected from the neighborhood cell library and added to the developing patch. The appropriate raster cells inside its neighborhood range are likewise added to the library of neighborhood cells.

(6) Patch Completion:

Steps (4) and (5) are continued until the patch is the size desired.

(7) Generating Multiple Patches:

Step (1) restores the patch seed cell library, and then steps (2) through (6) are repeated to construct the next patch. It should be noted that due to the influence of the newly formed patches, information about whether raster cells in the patch seed cell library have been developed or if they are connected to already developed raster cells may change at this point.

Furthermore, upon initiating a new simulation time step, MUSE only conducts the procedure stated in step (1). In other words, throughout each simulation time step, all urban growth patches use the same patch seed cell library.

Figure 5-2 depicts the SPGE patch growing process. We have set the patch position uncertainty parameter to 1, which means that only the method of optimizing patch urban development suitability is used.

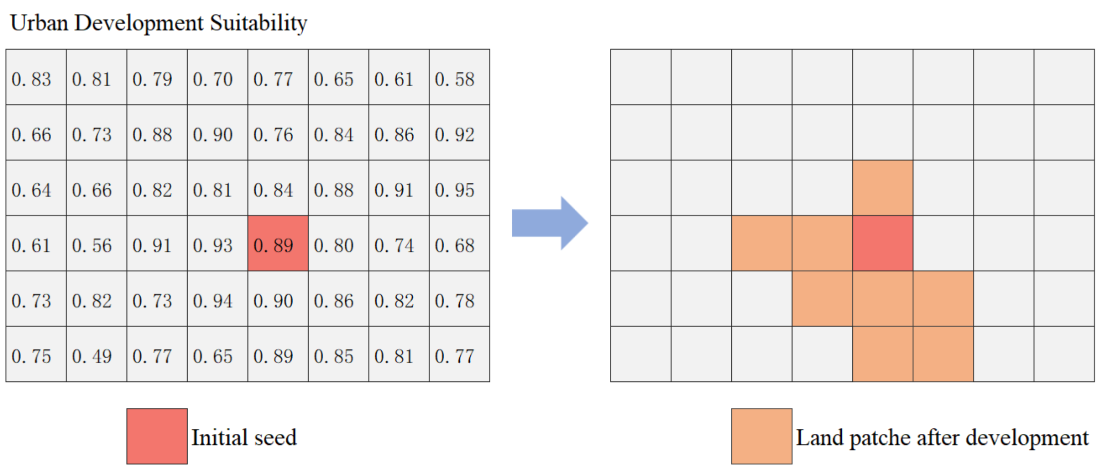



Figure 5-2: Schematic Illustration of SPGE Patch Growth

### 5.4.2 PPGE

The Parametric Patch Generation Engine (PPGE) adds additional techniques for manipulating patch shapes to the Simple Patch Generation Engine (SPGE). Users can now affect patch shapes in terms of compactness, continuity, and other properties thanks to this update. Furthermore, by assigning different weights, it achieves a compromise between enhancing patch urban development appropriateness and improving patch morphologies.

The PPGE's procedures closely resemble those of the SPGE. The fundamental difference, however, is how the PPGE adjusts the initial urban development suitability of raster cells inside the neighborhood cell library using a patch shape control technique that is dependent on input data. This modified development suitability criteria aids the PPGE in selecting and reviewing raster cells for potential inclusion in the intended generated patch, ensuring that the cells meet the changed development suitability criteria.

The method of adjusting urban development suitability is as follows:

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

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

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.

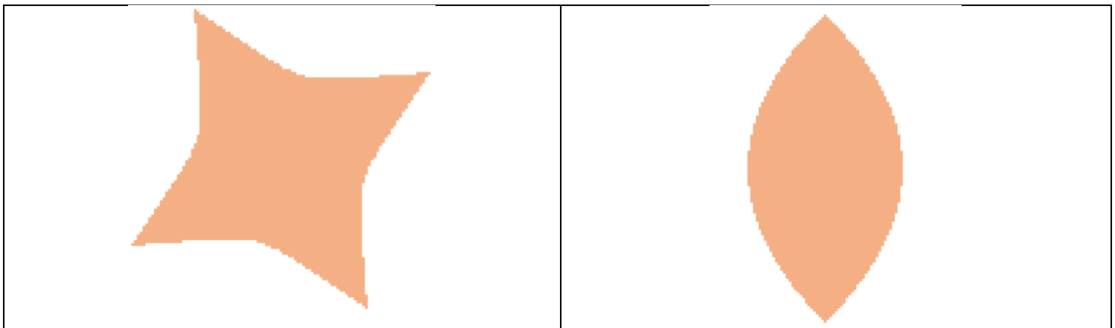
The implementation of the patch shape control algorithm is based on the following formula:

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

In this equation, the patch's shape control score ($\mathrm{Shape}_{\mathrm{score}}$) is determined by considering the distance ($\mathrm{dist}$) and direction ($\theta$, 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 ($\mathrm{N}$) and denominator ($\mathrm{D}$) control the ratio between the closest and farthest distances from the seed point to the boundary of the area. Parameter $\mathrm{A}$ defines the exponent of the patch, while parameter $\mathrm{O}$ determines the direction of the longest axis. The length of the bounding rectangle's diagonal ($\mathrm{diag}$) is used to normalize the shape control score.

Additionally, the patch generation engine provides 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 shifts to shape control. Conversely, when shape suitability is less than development suitability, the emphasis turns to increasing development suitability.

Figure 5-3 illustrates various patch shapes under different parameter settings. It's important to note that we adjusted the patch location uncertainty parameter to 1 during the patch growth process to emphasize shape differences, showing that we used the strategy that maximizes patch urban development adaptability. Please see the following publication for a more in-depth knowledge of the engine's functioning principles: https://doi.org/10.1080/136588197242329



N=1 D=2 A=4 O=30 shape_weight=1 suit_weight=0 | N=1 D=2 A=2 O=0 shape_weight=1 suit_weight=0

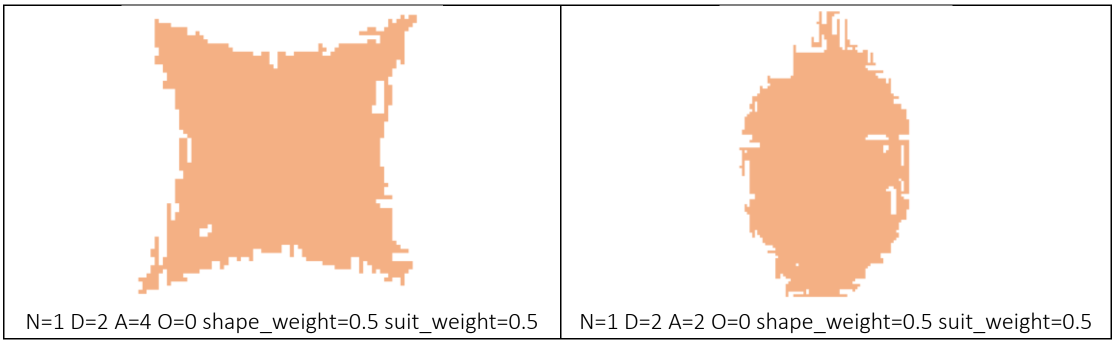



Figure 5-3 Schematic of PPGE Patch Growth

### *5.4.3 Nei-PGE*

The Neighborhood-Controlled Patch Generation Engine (Nei-PGE) is based on the idea of creating patches based on raster cells' urban development suitability. It also incorporates modifications to raster cell development appropriateness based on whether surrounding raster units were included in the patch generating process. This modification allows you greater control over patch shapes.

Specifically, after identifying the initial seed cell, suitable raster cells within the neighborhood range of the seed cell are gathered into a neighborhood cell library. This library serves as a source of candidate cells for the seed cell's patch expansion. When a newly added neighborhood cell is already present in the library, the likelihood of urban development adaptability of these duplicate neighborhood cells is multiplied by a decimal value beta in the range [0, +]. When beta is between 0 and 1, the patch shape becomes less compact. When beta is between [1, +], the patch shape tends to be more compact and round.

The urban development suitability for duplicated neighborhood cells is only adjusted once. The appropriateness of cells that have been added several times cannot be changed. Following that, based on the updated appropriateness, the patch creation engine selects raster cells from the neighborhood cell library. The subsequent procedures are the same as with the Simple Patch Generation Engine. Please see the following publication for a more in-depth knowledge of how this engine works: https://doi.org/10.1016/j.landurbplan.2022.104640.

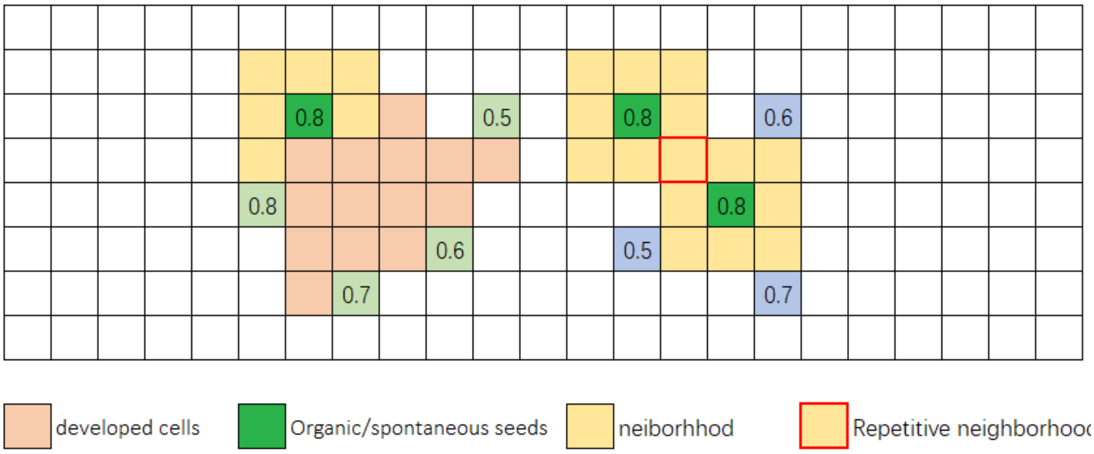



Figure 5-4 Schematic of organic/spontaneous seeds and their neighborhood

### *5.4.4 Dis-PGE*

The Distance-Controlled Patch Generation Engine (Dis-PGE) generates patches depending on raster cells' urban development appropriateness. At the same time, it uses the Euclidean distance between raster cells in the neighboring cell library and the patch's initial seed raster cell to alter the development appropriateness of these cells. This setting gives you control over the patch shapes. The following is the adjustment procedure:

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

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*

Modifying the percentage of organic growth patches in urban land can have a substantial impact on the layout's compactness and continuity. Higher percentage of organic growth patches encourage urban expansion that connects with existing patches, resulting in more continuous urban regions. Lower proportions of organic growth patches, on the other hand, tend to follow a self-generated growth pattern in which new patches do not interact with adjacent developed areas, resulting in isolated development zones. This urban growth pattern demonstrates a more pronounced expansive expansion state, resulting in a more dispersed overall form. Figure 5-5 depicts the distinct properties of these two groups.

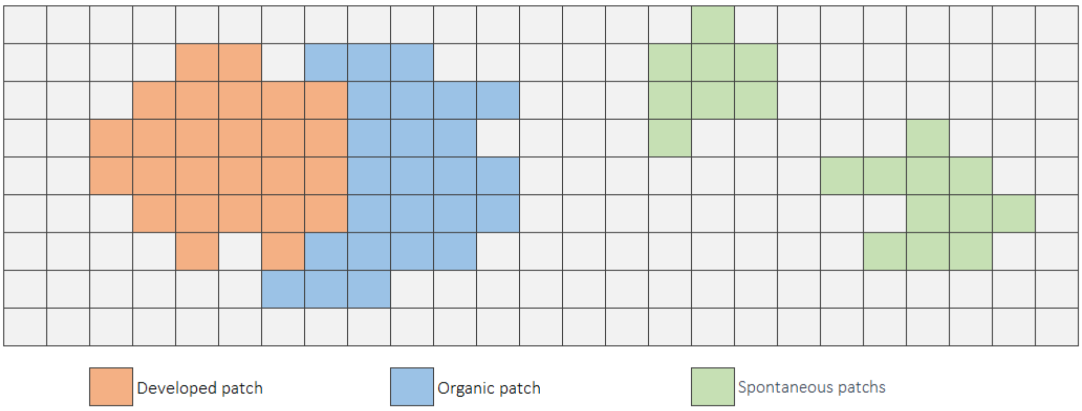



Figure 5-5 Schematic diagram of organic growth patches and spontaneous growth patches

### *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 selects cells in the candidate neighborhood region with the highest probability values. While this may improve model stability and repeatability to some extent, it may also cause the model to become stuck in local optima, making it sensitive to initial conditions and perhaps disregarding other possible development areas. Randomness is incorporated in the stochastic selection mode, making the selection process more flexible and diversified. By integrating randomness, the model can investigate a broader variety of possibilities in the candidate area, freeing itself from the confines of prior probability. This stochasticity can help to reduce the problem of local optima in deterministic selection by making the model more resistant to initial conditions and 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 affects the urban growth pattern's balance of deterministic and stochastic selections. By altering this parameter, one can control the proportion of deterministic and stochastic selections, influencing the model's overall behavior. When the parameter is increased, the model shifts toward deterministic selection mode, improving stability and predictability. A smaller parameter value, on the other hand, favors the stochastic selection mode, allowing the model to engage in more random exploration during patch selection.

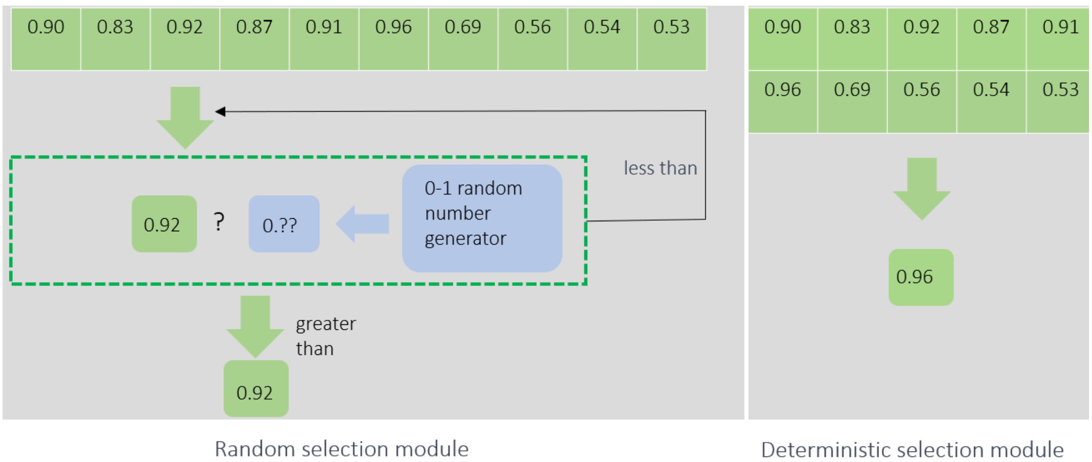



Figure 5-6 Schematic diagram of random selection and deterministic selection

### *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:

$$A = Len \cdot T \qquad (10)$$

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 Vesion 1.0

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

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

Contact infomation：

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

Copyright 2023 LandInsightLab All Rights Reserved