Methodology to generate landscape configurations for use in multi-actor plan-making processes

C.T.J. Slager, B. de Vries, A.K. Bregt[1] and A.J. Jessurun
Eindhoven University of Technology
[1] *Wageningen University*

P.O. Box 513
5600 MB Eindhoven
The Netherlands
c.t.j.slager@tue.nl

Key words: landscape configurations, spatial allocation, Landscape grammar, Cellular automata, Genetic Algorithms, Simulated Annealing

Abstract: In this paper, we investigate an approach to generate landscape configurations for use in multi-actor plan-making processes. Using the information from pre-defined lot typologies, a heuristic allocation method, consisting of a suitability function and an allocation mechanism of lot components is explained. The suitability function is primarily based on adjacency and distance parameters as found in landscape design literature. The allocation mechanism starts from a random but constrained initial situation, and generates a plausible lot configuration by orderly swapping pairs of cells thereby increasing the overall suitability of the plan . From the results, the limitations of this approach are concluded and the concepts are presented for an improved landscape generation algorithm.

1. INTRODUCTION

Strategic regional spatial plan-making processes in the Netherlands can be characterized as complex iterative search processes in which multiple actors with conflicting interests participate and try to develop a realistic integrated development vision for the area under consideration. (Däne and van den Brink, 2007) Traditionally, maps and derivatives play an important communicative and informative function in spatial planning. Not only providing ultimate information about various themes of the current state of the area, in spatial planning current topographic and land-use maps form the basis (as pad) for the design of a new development map for an area via the process of sketching and allocation of physical layout and land-use functions.

Nowadays, in collaborative plan-making processes, scenarios and finally the map is gradually developed starting from coarsely zoning the area on sketch paper with color outlined and hatched surfaces, to more precise delineating polygons (often snapping to existing structures) with consistent and uniform use of colors, hatch patterns and legend definitions. Dependant on the legal status of the plan (and its map) this preciseness should be perceived with care.

For many years, the scientific community develops separate digital tools to complete decision-support systems, referred as planning support systems (PSS) in order to assist plan-making participants, including decision-makers in their specific tasks(Geertman and Stillwell, 2004). Due to the substantial development in GIS-technology, most effort in specific relation to planning support has been put in quantitive analysis of the monitoring data and modeling, forecasting and evaluating potential developments of the future environment. Besides, substantial research has been focused on the 2D and 3D visualisation capacity of computers and how this can be applied in the communicative setting of spatial planning processes.

In this article we discuss the development of a tool to efficiently and consistently generate landscape configurations as a deepening of the current abstract colored and hatched pattern representation including related legend definition in analogue maps. As such, this generation provides a qualitative detailed representation of proposed development ideas and provide the desired detail level for quantitive evaluation models. The user interactively defines and assigns 2D and 3D landuse typologies and the system generates a plausible landscape configuration. This generation is considered a two-stage-procedure: 1) definition of the zone typology and allocation of the lot typologies, as defined in the zone typology and 2) allocation of the lot components, as defined in the lot typology. This article focuses on the stage 2 and proposes a method to spatial allocate lot components in a test lot, subject to a set of objectives.

2. RELATED RESEARCH

In the field of urban and landscape configuration, several methods can be discerned. We briefly introduce the basic concepts broadly used in this field to (re)produce landscape models.

2.1 Grammars – Landscape grammar

Mayall and Hall recently introduced the concepts (Mayall and Hall, 2005) and the implementation (Mayall and Hall, 2007) of a so-called landscape grammar, mainly influenced by pioneering shape grammar research of Stiny in the 1980s. As they state, landscapes "…constitute a (spatial) ordering and visual expression of objects in two dimensions (2D) and three dimensions (3D) that allow them to be read, written (created or modified), and understood by humans." (Mayall and Hall, 2005) One can identify an analytical and a constructive part when dealing with grammars. First, landscape objects that exist in a specific landscape scene (region of interest) are inventoried and saved in a vocabulary. Spatial and non-spatial relations between these objects are identified and expressed in a set of rules. This analytical process defines a landscape's character. In the constructive part of the grammar this knowledge is used to generate simulated landscape scenes. In a serial modeling process the landscape is iteratively (re-)built using an initial scene, additive placement and modification of the inventoried landscape objects and the firing of defined rules. Because of stochastic elements in the process, many generated scenes are likely to be different. The full range of generated scenes is called landscape language, and comprises the set of all scenes possible be generated by a given landscape grammar and its interpretation. (Mayall and Hall, 2005) Finally, planning regulations can be expressed in the grammatical format and used to influence the character of the landscape and thereby giving the possibility to illustrate effects of intended spatial plans.

Lots of spatial grammars are introduced recent decades, however in this field of research, main challenges remain considering the feasibility and flexibility and therefore practical utility of the developed grammars.

2.2 Cellular automata models (CA)

Probably, the most well-known approach in the field of landuse/landcover modeling, Cellular Automata (CA) models are able to generate complex spatial structures based on relatively simple set of rules.

The original concepts are introduced by von Neumann and Ulam in the 1940s in order to investigate and model underlying processes of life (Jiao and Boerboom, 2006). CA in that respect is very suitable and also often used for modeling natural phenomena. CA in specialized forms, have been used in many fields of research in which space-time modeling is apparent. In its fundamental form it consists of five main components (Wolfram, 1984; Itami, 1994):

i) **Lattice**: abstract gridcell-representation of the space to be modelled;

ii) **Cell state**: a value which a cell can take and which represents the state of the phenomena in question; in its original form this is of Boolean type: 0 or 1, in other forms its often implemented as integer type (e.g. landuse), but can be of any possible data type;

iii) **Neighbourhood**: the state of the examined cell is influenced by cell states of the neighbours, which is defined by the neighborhood. In general, a distinction is made between use of the von Neumann (four adjacent cells) or the Moore (four adjacent plus four diagonally adjacent cells) neighbourhood.

iv) **Transition rule**: this is the control component of the CA; it determines if and how the examined cell state will change state in the next time iteration as a function of its neighbourhood

v) **Time**: discrete time steps representing time in the model

Most CA-research has been conducted on the definition of transition rules, since it determines the final behavior of the model (Jiao and Boerboom, 2006). In landscape modelling CA is often used for scenario studies investigating developments of urban areas given certain kinds of constraints, e.g. spatial, socio-economical criteria (Lau and Kam, 2005; Stevens and Dragicevic, 2007), besides landuse/landcover change modelling (Parker et al., 2003). In its fundamental form, CA is not necessarily an optimization algorithm, like genetic algorithms or simulated annealing.

2.3 Evolutionary models – Genetic algorithms (GA)

Genetic algorithms, introduced by Holland (Holland, 1975), are efficient search algorithms, which generate (near-)optimal solutions for optimization problems (Goldberg, 1989), primarily based on the Darwinian theory of natural selection and genetics (Zhang and Armstrong, 2008). They are particularly useful when a solution has to be optimized for more than one objective, also known as combinatorial optimization.

In its fundamental form, the method tries to generate, from a population of individuals (solutions), a satisfying solution in an evolutionary manner.

The process traditionally starts with a random initialized population of solutions. From this population the best-ranking ('fittest') individuals are stochastically selected and possibly adapted (using crossover or mutation) into a new population. Ranking is based on the fitness each individual has with respect to a set of objective functions. The new population is used in the next generation (iteration), until a termination condition, e.g. satisfaction of criteria or fixed number of iterations, is reached.

GA is widely used, partially or completely, in many different research disciplines, like in spatial planning,. (Feng and Lin, 1999; Loonen et al., 2007) It has proven to be very useful in generating sets of near-optimal (urban) designs or resource allocation. The choice of representation and the exact interpretation of the algorithm have a large influence on the generated final results.

2.4 Simulated annealing (SA)

Simulated annealing, introduced by Kirkpatrick et al. in 1983 (Kirkpatrick et al., 1983), is in a certain way comparable with genetic algorithms, since it is useful as combinatorial optimisation, as well. In contrast with genetic algorithms, simulated annealing imitates the physical process of crystallisation. (Duh and Brown, 2007) Simulated annealing can be seen as a certain GA, but with one individual and only mutation as genetic operator. Simulated annealing starts with an initial situation. This initial situation, often represented as a grid of cells, has to be optimized for an (set of) objective(s), and holds an 'energy level', comparable with a fitness value in GA. After a random swap of cells, the new situation is tested for change of its 'energy level'. The state change is not only accepted if the 'energy level' is smaller (fitness is higher) than the previous situation, if 'energy level' is higher (fitness is lower) the state change is accepted with a certain probability, as well. This is implemented to escape local minimum solutions.

The probability of acceptance is given by the Metropolis criterion (Aerts and Heuvelink, 2002):

$$P(acceptchange) = \exp\frac{f_0 - f_1}{s_0}$$

where f_0 and f_1 are the 'energy levels' for the compared two situations and s_0 is the so-called 'freezing parameter', which is gradually decreased for each pre-defined number of iterations. This means that jumping to a higher energy becomes less and less likely towards the end of the iteration procedure

(Levine 1999). Another concept in simulated annealing is the ‘cooling schedule’, that consists of three important parameters of this method (Aerts and Heuvelink, 2002):
i) iteration length for each decrease of freezing parameter
ii) initial value of the freezing parameter s_0
iii) freezing parameter decrease factor

Although it can not be proven that simulated annealing guarantees an optimal solution, practice has shown that a sufficiently slow decrease of the freezing parameter yields in almost all cases the optimal solution. (Aerts and Heuvelink, 2002) Like GA, SA is an optimization method and gives in specific situations quicker and better near-optimal results than GA, since it uses a different procedure to find the solutions and correct for local minimum solutions.

3. METHODOLOGY

In this section, we start with explaining the concepts of the use of land-use typologies in the regional plan-making process. Next, we present the model that we is implemented to generate landscape configurations at the lot level.

3.1 Regional plan design and typologies

In this research the use of generic land-use typologies in collaborative plan-making is essential. These land-use typologies are the building blocks for the planning process, and the input parameters for the generation algorithm. This planning process intends to follow traditional procedures in collaborative plan-making as closely as possible. In addition, actors (e.g. planners, urban designers, project developers, etc.) are offered digital instruments in order to develop scenarios more efficiently and consistently and in such a way it is useful as input for extensive evaluation with existing GIS-models. (Slager et al., 2007) The lot typology is a central concept in this article and is a composition of a set of lot components; e.g. buildings, trees, grass, agricultural fiels, waterpool and parcel infrastructure. Each lot typology is distinctive from the other due to differences in presence and quantity of lot components and in position relative to other lot components. To test the concepts, four different and essential example generic lot typologies (20x20 cells) were designed to be able to focus on the main aspects of this component allocation and positioning. (figure 1) The typologies are oriented to an imaginary road in the south. With this set of example lot typologies, a substantial part of urban-rural landscape complexity existing in the Netherlands is covered, already.

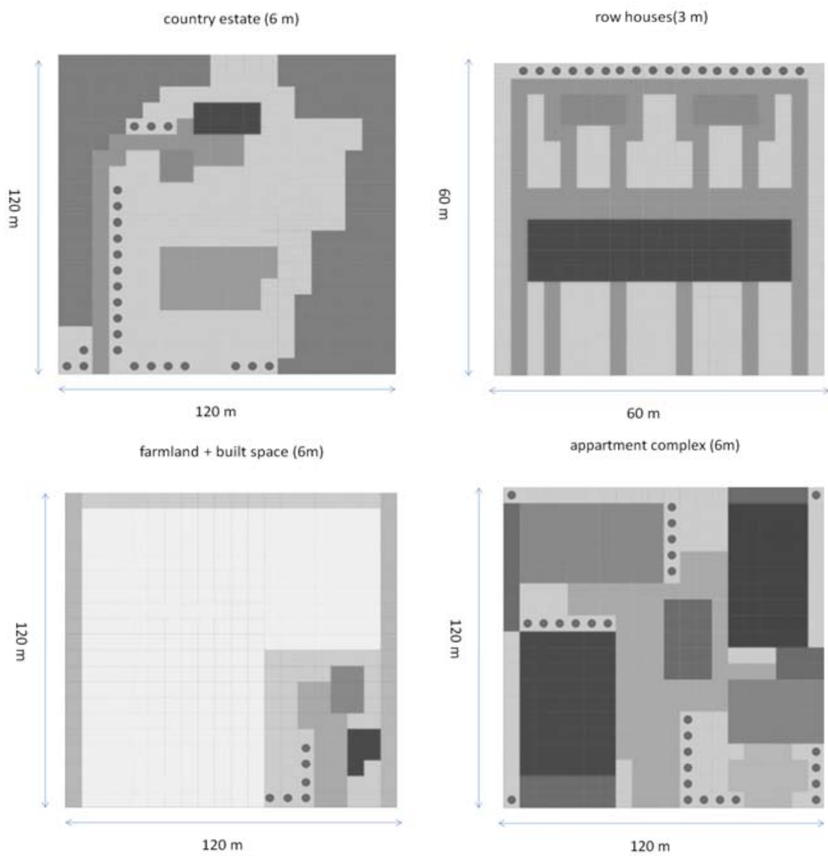

Figure 1. Four lot typologies to test the concepts. The *country estate* (top left) uses a minimum cell size of 6 m. Seven lot components can be discerned (from light to dark, respectively): 1) grass (> 1 instances); 2) path; 3) pond; 4) shed; 5) forest (> 1); 6) villa and 7) tree trunk (> 1). The *row houses* (top right) uses a minimum cell size of 3 m. Five lot components can be discerned (all with more than one instance): 1) grass; 2) path; 3) shed; 4) block of row houses; 5) tree trunk (>1). The *farmland incl. built space* (bottom left) uses a minimum cell size of 6 m. Seven lot components can be discerned: 1) maize; 2) grass (>1); 3) path; 4) ditch (>1); 5) barn; 6) farmhouse; 7) tree trunk (>1). The *appartment complex* (bottom right) uses a minimum cell size of 3 m. Seven lot components can be discerned: 1) grass (>1); 2) pond; 3) square; 4) parking; 5) app.complex (also 2 in black); 6) boskage (>1); 7) tree trunk (>1). Each component covers a certain indicative and relative area and can be topologically defined. It should be emphasized that these lot typologies are compositions; final ordering of components and sizes may be liable to stochasticity and local factors.

3.2 Design considerations

The lot typologies indicated in figure 1 have some implicit ordering of lot components. The developed algorithm should strive for generating lot configurations as shown in the figure. It should emphasized here that the typologies rather serve as a generic composition of the proposed landscape design; this means that defined components should be allocated, however plausible variety with respect to component shapes and its final position in the lot are allowed, since lot border layout and its neighbourhood puts constraints as well. As can be observed from the typologies, cell size is dependant on lot typology.

There should be a distinction in approaches for generating object-type components and network-type lot components (e.g. slip roads, driveways). Network-type lot components and its place in allocation sequence will be treated in a later stadium of the research. A main consideration is that the position of the component is related with adjacencies and distances with other lot components.

3.3 Representation of the test lot

We first examined the usability of a heuristic method for allocation of area-type land-uses developed for a higher abstraction level, proposed by (Arentze et al., 2006). Their proposed method consists of a suitability function and an allocation mechanism. The method is adjusted to our specific situation. Before we treat these concepts, we first explain the way the test lot is represented. The test lot is represented by a raster of grid cells (20x20 cells). Each grid cell (i) holds information about coordinates in the test lot (i_{xy}), has an attribute in which the lot component value (i_v) is saved and finally has an attribute in which can be stated if this component value of the cell is *fixed* during the process (i_f), i.e. unable to be converted to some other value. A cell is set fixed to be able to include testing of effects of a neighbouring lot and its components on the allocation inside the test lot. In the initial situation, i_v of each cell in the test lot is set the value *null*, i.e. has not a value yet.

3.4 Suitability function

The suitability function in the original method is based on a function proposed by Engelen et al. 1997. They suggest that the suitability or potential of a cell for a particular land-use at that location depends on land characteristics of the cell, on distances, and on adjacencies to each type of land-use (Arentze et al., 2006). In our particular case, this concept is not applied at the land-use level (e.g. housing, retail, parks etc.) but at the level of the landscape components. A cell with a component value should in this case be considered as *a part* of a landscape component. A landscape component could then be considered a cluster of cells ('parts') with equal values. For each landscape component, the suitability is defined by (Arentze et al., 2006):

$$S_{ij} = \sum_{k} X_{ik}^{j} + \sum_{j'} D_{ij'}^{j} + \sum_{j'} A_{ij'}^{j}$$

where S_{ij} is the suitability score of cell *i* for lot component *j*, X_{ik}^{j} represents the (weighted) score of land characteristic *k* of cell *i* regarding lot component *j*, $D_{ij'}^{j}$ is the score of the distance class representing the shortest distance from cell *i* to the nearest cell with lot component *j'* for lot component *j* and $A_{ij'}^{j}$ is the score of the adjacency to lot component *j'* for lot component *j* in cell *i*. In this article, land characteristics of a cell are not taken into consideration. For each instance of a lot component a distance and adjacency table should be made. Examples of D and A tables are shown in Table 1 and 2. Corresponding distance neighbourhoods are visualized in figure 2.

Distance in this case is expressed in cell size, cell size is variable for each test lot and partially dependent on related lot typology. The component based distance covers possible interactions between lot components, while the adjacency score represents the weighted sum of effects across neighbouring cells.

Table 1. Adjacency table for farmhouse; each iteration the contribution of adjacency to the suitability of an examined cell is calculated using the Moore neighbourhood (figure 2). If the examined cell has the value farmhouse; the score is calculated with figures from this table.

Farmhouse		Neighbourhood	
	A	N/S/E/W	NW/SW/SE/NE
Lot component	grass	0	0
	maize	0	0
	barn	0	0
	farmhouse	1	1

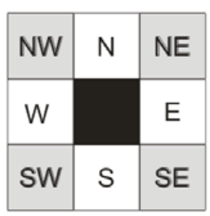

Figure 2. Moore neighbourhood used in adjacency score calculation

Table 2. Distance table for farmhouse; each iteration the contribution of distance of an examined cell is calculated using an extended neighbourhood (figure 3). If the examined cell has the value farmhouse; the score is calculated with figures from this table.

Farmhouse		Neighbourhood								
	D	1	2	4	5	8	9	10	13	18
Lot component	grass	1	1	0	0	0	0	0	0	0
	maize	-1	-1	0	0	0	0	0	0	0
	barn	0	0	0	0	0	3	3	1	1

18	13	10	9	10	13	18
13	8	5	4	5	8	13
10	5	2	1	2	5	10
9	4	1		1	4	9
10	5	2	1	2	5	10
13	8	5	4	5	8	13
18	13	10	9	10	13	18

Figure 3. Extended neighbourhood used for distance score calculation

3.5 Allocation mechanism

For spatial allocation of the lot components the following procedure is suggested (Arentze et al., 2006). The chosen lot typology T_j defines the number and percentage of lot components to be realized in the test lot,
$j = 1 \dots J$. This method assumes that solutions in any stage of the generation process are consistent in terms of component area as given in $T_{j.}$ The initial situation (for which the lot component value i_v for each cell is *null*) is randomly filled according to the percentages given for each lot component in T_j. This first allocation ensures consistency in terms of the basic lot type, but remains far from an acceptable solution, since no relational information between components is processed so far.

Therefore, the system initiates a global optimization procedure in which the total utility of a current solution is calculated as follows:

$$U = \sum_i \sum_j S_{ij} \times a_{ij}$$

where U is the utility of the current solution and $a_{ij} = 1$, if lot component j has been assigned to cell i in the adapted situation, and $a'_{ij} = 0$, otherwise. The system starts swapping pairs of different lot components across the test lot, to improve the utility of the solution. After a swap the total utility value of the adapted situation (U') is calculated. If the utility of the adapted situation is equal or larger than the current situation (i.e. $U' >= U$), the current situation is set to the adapted situation. Else, the swap is reset. We used a heuristic to select the cell pairs to be swapped in sequence. The test lot can be considered a set (x,y):

$$\{(x, y); x \in C \cap y \in C : x < y\}$$

where *C* is the set of cells in the test lot. Using the coordinates of the cells, the cells in the test lot can be considered as ordered row-wise. The system uses row-wise processing in order to select pairs of cells to be swapped.

The system continues swapping (even in new loops) until no more swaps can be made to increase the utility of the current solution. The allocation of the lot components in the test lot is then completed. In figure 4 a current situation and an adapted situation (figure 5) are simulated, with the assumption that only adjacency values are taken into account. The same procedure is applicable to calculate distance scores. In this example the total utility of the adapted situation (U' = 2) is not equal or better than the utility of the current situation (U = 4). This means that the swap is reset and the next pair of cells (x_{00} and y_{02}) are applied and tested.

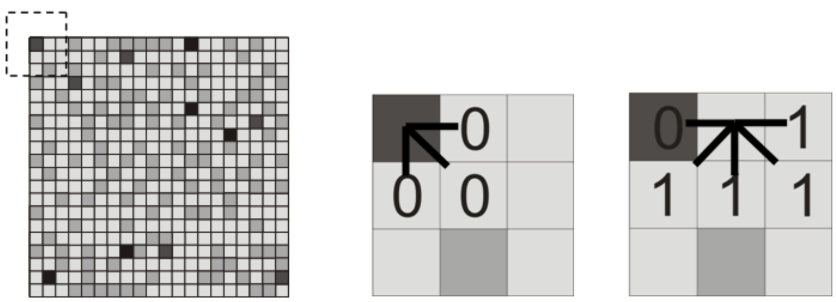

Figure 4. Visualisation of current situation (prior to swap). The test lot comprises of only four of the lot components corresponding to the typology *farmland incl. built space* (from light to dark, respectively): 1) maize; 2) grass; 3) barn; 4) farmhouse. The utility of the current situation U = (0 + 0 + 0) + (1 + 1 + 1 + 1 + 0) = 4

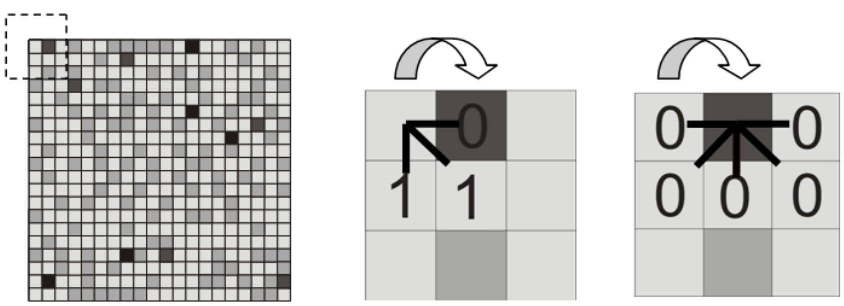

Figure 5. Visualisation of adapted situation (after swap). The utility of the adapted situation *U'* = (0 + 1 + 1) + (0 + 0 + 0 + 0 + 0) = 2

4. RESULTS

The described method is implemented in the high-definition programming language Ruby and it takes on average 100 seconds computation time on a Intel core 2 CPU T7200 2.00 GHz and 2 gb RAM, to perform one run for a grid of 20x20 cells and a distance grid of 3x3. If the grid is extended the time needed explodes exponentially. The method produces deterministically (provided an identical initial situation) clusters of cells in an acceptable time period for relatively small grids (for an example see figure 6), however the final solutions are not completely satisfactory, mainly because of:

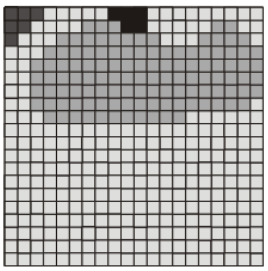

Figure 6. Visualisation of end result after optimalization method

i) determining the values for the adjacency and distance tables is not a straightforward activity, since the amount of tables to be filled in for each lot component could be very large; moreover the landscape design solutions aimed for cannot always be generated, since the table values are not transparently correlated;
ii) the minimum desired distance between lot components cannot be set due to the fact the algorithm does not treat a formed cluster as a lot component but it treats each cell individually;
iii) swapping of cells via row-wise processing has an enormous (deterministic) influence on the yielded solution;
iv) the method used is not very time-efficient; in spite of several opportunities to optimize the method, calculation of distance scores is the most costly operation; distance grid size will therefore always be the limiting time factor.

5. DISCUSSION

In this article we investigated a spatial allocation method which generates a plausible test lot configuration. The proportion of the physical lot components are defined in the lot typology. The results suggest that the proposed methodology is a good start in search for a method to generate plausible landscape configurations. Yet, several important lessons can be learnt.

With respect to broadly used methods in this field, concepts of each treated method are recognized in the method that we adopted from Arentze (2006). Comparably, the lattice, cell state and neighbourhood defined in CA are also feasible here. Additionally, the neighbourhood is extended, the transition rule is expressed as a score calculation and complete cell state evaluation for each time step, is substituted for an ordered swap of cells. Not all cell states are evaluated each iteration (time step), only the swapped cells are examined. In this respect, the method mainly resembles a stripped version of simulated annealing. The swapping mechanism is comparable, however simulated annealing assumes complete random swapping of cell pairs (Aerts and Heuvelink, 2002). Furthermore, the freezing parameter and cooling schedule has been neglected, so far. Based on the presented results and existing research on landscape configuration and spatial allocation, we aim for a more sophisticated method in which we can clearly state (defined objectives) *how many* instances of each component we like to generate, *how large* and *how compact* each should be and finally *how* each is *located* in relation to each other component.

We consider the representation of the lot typologies and the test lot as feasible for further implementation. Unlike the knowledge-informed generation of initial situations in landscape grammar and some CA procedures, but in concordance with the described genetic algorithm and simulated annealing, randomly generated initial situations are found usefull since the effects of changing transition rules can be fully investigated. Besides, it assures that in each phase of the solution generation the proportions apply. Further usage of the distance and adjacency score tables seem unlikely, since actual landscape patterns and configuration cannot fully be expressed and predicted. These tables have most resemblance with the transition rules of CA and are likely more useful in generating urban patterns which are influenced by human behaviour modeling, like in (Arentze et al., 2006). In further implementation we substitute the suitability function and continue with the allocation mechanism. Selection procedures in the cell swapping algorithm will be substituted by a complete random selection procedure in order to remove the strong identified influence of the swapping procedure on the results and to optimize computation time, as well.

6. OUTLOOK

Due to considerable time inefficiency of Ruby code execution, it is decided to change to the Java programming language, which also delivers a substantial amount of existing libraries. After implementation of the objectives (e.g. certain amount, area distribution, compactness and location of the components), the four primary lot typologies will be tested for plausible allocation in the test lot. To test if the results are considered plausible, experts in the field of spatial planning will validate the results in two approaches; 1) show the results and use of a questionnaire about how the allocated lot components are situated in respect to its neighbourhood and if it represents the expectation of the user and 2) not asking if the result is plausible, but provide easy-to-use tools to adjust the plan configurations manually, and measure the differences between generated and adapted plan. We hope to report on these plausibility experiments in the near future.

7. REFERENCES

Aerts, J. C. J. H. and G. B. M. Heuvelink, 2002, "Using simulated annealing for resource allocation", *International Journal of Geographic Information Science*, 16(6), p. 571-587.

Arentze, T. A., A. W. J. Borgers and H. J. P. Timmermans, 2006, "A heuristic method for land-use plan generation in planning support systems", in: J. P. van Leeuwen and H. J. P. Timmermans (eds.) *Progress in Design & Decision Support Systems in Architecture and Urban Planning, Eindhoven, The Netherlands*, Springer, Dordrecht, p. 135-152.

Däne, S. and A. van den Brink, 2007, "Perspectives on citizen participation in spatial planning in Europe", in: A. van den Brink, R. J. A. van Lammeren, R. van de Velde and S. Däne (eds.) *Geo-visualisation for participatory spatial planning in Europe: Imaging the future*, Wageningen Academic Publishers, Wageningen, p. 33-50.

Duh, J.-D. and D. G. Brown, 2007, "Knowledge-informed Pareto simulated annealing for multi-objective spatial allocation", *Computers, Environment and Urban Systems*, 31, p. 253-281.

Feng, C.-H. and J.-J. Lin, 1999, "Using a genetic algorithm to generate alternative sketch maps for urban planning", *Computers, Environment and Urban Systems*, 23, p. 91-108.

Geertman, S. and J. Stillwell, 2004, "Planning support systems: an inventory of current practice", *Computers, Environment and Urban Systems*, 28, p. 291-310.

Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Itami, R. M., 1994, "Simulating spatial dynamics: cellular automata theory", *Landscape and Urban Planning*, 30, p. 27-47.

Jiao, J. and L. Boerboom, 2006, "Transition Rule Elicitation Methods for Urban Cellular Automata Models", in: J. P. van Leeuwen and H. J. P. Timmermans (eds.) *Innovations in Design & Decision Support Systems in Architecture and Urban Planning*, Springer, Dordrecht, p. 53-68.

Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi, 1983, "Optimization by Simulated Annealing", *Science*, 220(4598), p. 671-680.

Lau, K. H. and B. H. Kam, 2005, "A cellular automata model for urban land-use simulation", *Environment and Planning B: Planning and Design*, 32, p. 247-263.

Loonen, W., P. S. C. Heuberger, A. H. Bakema and P. Schot, 2007, "Improving the spatial coherence of nature areas using genetic algorithms", *Environment and Planning B: Planning and Design*, 34, p. 369-378.

Mayall, K. and G. B. Hall, 2005, "Landscape Grammar 1: spatial grammar theory and landscape planning", *Environment and Planning B: Planning and Design*, 32, p. 895 - 920.

Mayall, K. and G. B. Hall, 2007, "Landscape Grammar 2: implementation", *Environment and Planning B: Planning and Design*, 34, p. 28-49.

Parker, D. C., S. M. Manson, M. A. Janssen, M. J. Hoffmann and P. Deadman, 2003, "Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review", *Annals of the Association of American Geographers*, 93(2), p. 314-337.

Slager, C. T. J., A. Ligtenberg, B. Vries de and R. S. Waard de, 2007, "Simlandscape: serious gaming in participatory spatial planning", in: M. Wachowicz and B. L. (eds.) *Proceedings of 10th AGILE International conference on Geographic Information Science. The European Information Society: Leading the way with Geoinformation (cd-rom). Aalbork, Denmark*, p.

Stevens, D. and S. Dragicevic, 2007, "A GIS-based irregular cellular automata model of land-use change", *Environment and Planning B: Planning and Design*, 34, p. 708-724.

Wolfram, S., 1984, "Cellular automata as models of complexity", *Nature*, 31(4), p. 419-424.

Zhang, X. and M. P. Armstrong, 2008, "Genetic algorithms and the corridor location problem: multiple objectives and alternative solutions", *Environment and Planning B: Planning and Design*, 35, p. 148-168.