Genetic Programming for Computationally Efficient Land Use Allocation Optimization

Moritz J. Hildemann[1]
Institute for Geoinformatics, University of Münster, Germany

Alan T. Murray
Department of Geography, University of California at Santa Barbara, CA, USA

Judith A. Verstegen
Department of Human Geography and Spatial Planning, Utrecht University, The Netherlands

Abstract

Land use allocation optimization is essential to identify ideal landscape compositions for the future. However, due to the solution encoding, standard land use allocation algorithms cannot cope with large land use allocation problems. Solutions are encoded as sequences of elements, in which each element represents a land unit or a group of land units. As a consequence, computation times increase with every additional land unit. We present an alternative solution encoding: functions describing a variable in space. Function encoding yields the potential to evolve solutions detached from individual land units and evolve fields representing the landscape as a single object. In this study, we use a genetic programming algorithm to evolve functions representing continuous fields, which we then map to nominal land use maps. We compare the scalability of the new approach with the scalability of two state-of-the-art algorithms with standard encoding. We perform the benchmark on one raster and one vector land use allocation problem with multiple objectives and constraints, with ten problem sizes each. The results prove that the run times increase exponentially with the problem size for standard encoding schemes, while the increase is linear with genetic programming. Genetic programming was up to 722 times faster than the benchmark algorithm. The improvement in computation time does not reduce the algorithm performance in finding optimal solutions; often, it even increases. We conclude that evolving functions enables more efficient land use allocation planning and yields much potential for other spatial optimization applications.

2012 ACM Subject Classification Computing methodologies → Discrete space search

Keywords and phrases Land use planning, Spatial optimization, Solution encoding, Computation time reduction

Digital Object Identifier 10.4230/LIPIcs.GIScience.2023.4

Supplementary Material *Software (Source Code)*: `https://doi.org/10.17632/4tw223jvjv.3`
Dataset (Illustrations): `https://doi.org/10.6084/m9.figshare.21977228.v2`

1 Introduction

Land is scarce, and the competition for land is increasing [4] and continues to increase in the future. Efficient planning can serve social, economic and ecological needs at the same time [4]. In contrast, inefficient and inconsiderate planning has much potential to cause future problems [15]. One aspect of land use planning is the allocation of land use activities.

In order to efficiently allocate land uses, land use planners specify the land use context by defining the land units, the land use categories, the scale, the benefits and undesired outcomes associated with the activities of future land use allocation. Land use modellers can translate these specifications into a solvable model: a land use allocation problem. The

[1] Corresponding author

modeller has to define the decision variable, the constraints, and the objective functions. The decision variable of the optimization is what land use category is assigned to which land unit. Land units can vary in their spatial representation, i.e. vector or raster, and in their encoding scheme.

Currently, the solution scheme in land use allocation optimization is a linear sequence of elements in which every element is one decision variable and is assigned one land use category. One encoding scheme is associating one land use category with one element in the sequence representing one land unit [2, 26]. Another option is to combine multiple neighbouring land units into patches and associate each patch with one element in the sequence [22, 29]. Then, benefits and undesired outcomes are formulated as objective and constraint functions. Constraint functions validate whether a solution violates the defined constraint(s), and objective function(s) quantify the solution's expected benefits.

Optimization algorithms identify solutions to the land use allocation problem. The problem specification determines whether exact algorithms are applicable to solve the problem or whether heuristic approaches are required. If the effort for solving the problem increases exponentially with the number of decision variables, the problem is NP-hard, and heuristic optimization methods are used [27]. Most land use allocation problems fall into the category of NP-hard problems: The number of land units u and the number of land uses categories luc defines the number of possible combinations n of the land use allocation problem: $n = luc^u$. Land use allocation algorithms using the standard encoding are slow when landscapes are complex [28], and face exponentially increasing computation times with increasing problem sizes [27].

Another encoding scheme, yet uncommon in spatial optimization, is a tree that organizes the elements recursively [24]. Since the choice of a suitable encoding has been proven to improve optimization [12] and land use allocation optimization encounters scaling problems with increasing numbers of land units, we propose using the recursive tree encoding. Trees can represent functions, and functions can represent fields [14]: If a function contains two variables, it is possible to represent continuous fields with longitude, latitude, and a variable. Therefore, the tree representation offers an alternative solution encoding scheme to represent spatial objects. Functions describe spatial patterns in the field of geostatistics [23], why should it not be possible to evolve continuous fields as functions to produce favourable land use patterns, for example, patterns that involve spatial compactness, or specific shapes of contiguous land uses?

Much research has been conducted to improve land use allocations with the standard solution encoding, but none on evolving functions to generate land use maps. This study aims to fill the research gap by opening the research domain to using functions as solution encoding. We propose a new method to map functions to nominal land use maps. We compare the new approach with state-of-the-art allocation algorithms on two multi-objective land use allocation problems, one raster and one vector land use allocation problem. In the remainder of this work, we are going to answer the following research questions:

1. How does the computation time of optimizing land use maps represented as functions scale with an increasing number of land units?

2. How does the function-evolving algorithm perform in comparison to state-of-the-art land use allocation algorithms in terms of computation time and the optimal solution quality?

2 Background

Heuristic search algorithms are most often used to solve land use allocation problems [26]. Heuristic search algorithms identify solutions that are not guaranteed to be truly optimal but help find "good enough solutions" for hard problems in finite time [27]. In contrast to exact optimization algorithms, heuristic optimization algorithms explore the search space of possible solutions until reaching a termination criterion [21].

Common heuristic optimization algorithms for solving land use allocation problems are population-based algorithms, e.g., Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO). GAs and single-objective PSO are commonly applied [26] for single-objective land use allocation problems, and the Non-dominated Sorting Genetic Algorithm II (NSGA 2) [6] is the most often used algorithm for solving multi-objective land use allocation problems [26]. Its successor, the NSGA 3 algorithm, leads to better distributed optimal solutions between conflicting objectives [18]. These algorithms use different search strategies: Genetic Algorithms mimic evolutionary processes by utilizing fitness proportionate selection and genetic recombinations of individuals within a population [10]. In PSO, the equivalent of an individual in a population is a particle in a swarm of particles, moving within the problem space [17] to find the best positions.

The algorithms evolve solutions by manipulating the solution. In land use allocation algorithms, the manipulation procedures of the algorithms are either applied on the sequences containing single land units [1] or of land use patches [16]. One advantage of using patches in comparison to single land units is the lower number of decision variables [29]. Another advantage of evolving patches is the higher likelihood of obtaining solutions with innate spatial relationships like adjacency or connectivity [17], which are often desired characteristics in land use allocation [26]. Numerous operators have been developed to steer the optimization process towards patches with certain characteristics such as compactness [17], or validity [29]. It is important to notice that some manipulations are computationally more demanding than others, but all manipulations of solutions with the common land use map encoding lead to an increased computational effort when considering more land units.

On the other hand, genetic programming (GP) is an evolutionary algorithm that evolves solutions with a different encoding: program trees that build functions [13]. The encoding yields the potential to evolve solutions detached from single land units by evolving fields that represent the whole landscape as one object: If the functions incorporate spatial variables, e.g. the latitude and longitude, the function produces an output variable for any given position. Other components of the function cans influence the output variable. Combined, the spatial and non-spatial components define how the variable varies in space. Therefore, it is possible to optimize the spatial variation of the output variable by manipulating the non-spatial components. Since the output variable is detached from land units, the number of land units does not affect the computational effort when manipulating the solutions.

The algorithm has been applied to a wide variety of non-spatial problems [20], but neither to land use allocation problems nor to spatial optimization problems in general. GP yields better results than GA in related applications, e.g. for generating grids of a continuous variable for photomosaics [19]. One identified reason is the higher flexibility due to the encoding of solutions, where little adaptions of the program trees can lead to many changes in the produced grid and potentially towards favourable patterns [19]. In addition to producing optimal grids of a continuous variable, genetic programming also proved to perform well on discrete variable classification [11]. The promising results of these studies suggest that genetic programming is applicable to allocating land use.

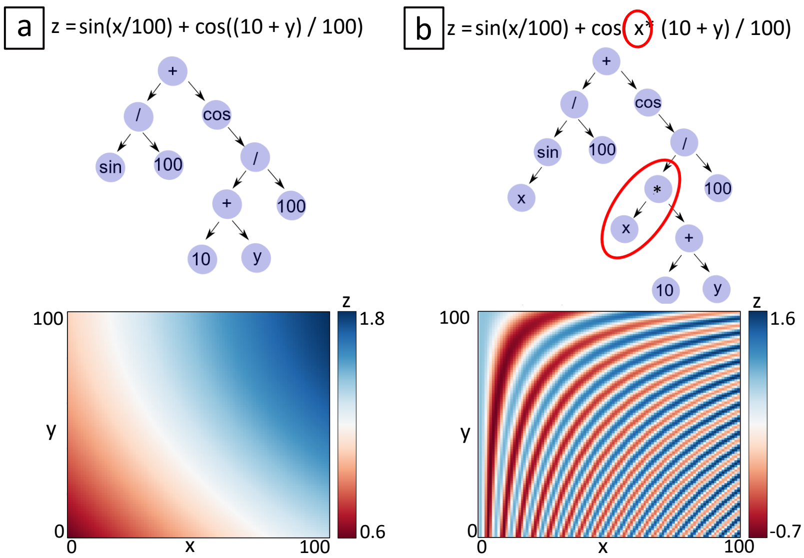

Figure 1 Two exemplary individuals with the function, the program tree, and the resulting field. The primitives are sine, cosine, addition, subtraction, multiplication, and division. The terminals are 10 and 100, and the x and y inputs range from 1 to 100, resulting in a continuous z value.

3 Methods

3.1 Land use allocation optimization using genetic programming

Generating fields with genetic programming

In genetic programming, every individual of the population is a "hierarchical composition of primitive functions and terminals" [13]. Typically, arithmetic operations, mathematical functions, or conditional logical operations constitute the functions [13]. The terminals and numeric constants are inputs to the problem. In our case, where solutions to the problems are two-dimensional fields, the inputs are x and y coordinates. The coordinates are two input variables that can repeatedly appear in the program trees (Fig. 1). When incorporating the coordinates within into mathematical functions, spatial For illustration purposes, the individuals are visualized as program trees (Fig. 1).

Mapping continuous fields to nominal land use maps

First, we retrieve the input coordinates from the land units. In the case of a raster representation of land uses, the row and column IDs serve as the x and y inputs of the program trees. In the case of a vector land use representation, the centroid coordinates of the land units serve as x and y inputs. Applying the function on the x and y inputs of the program trees defines the output variable z (Fig. 1). We use the mean of z per patch for the patch representation. Then z is min-max normalized to a range that matches the land use categories. Finally, rounding the normalized z-values to integers generates the desired nominal values.

This mapping procedure suffices to retrieve nominal values per land unit. However, the continuous variable contains an order, and mapping the continuous variable to a nominal variable propagates an order. It is not particularly meaningful to define an order between land uses urban, forest, or pasture. If this order were ignored, then the likelihood of neighbouring land uses would be influenced by the predefined land use order. To avoid this artifact, we

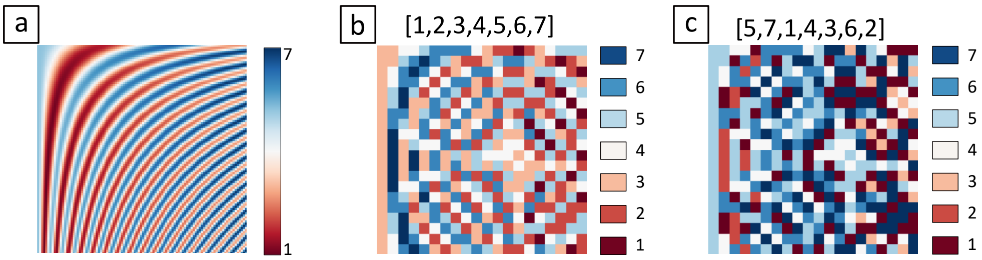

Figure 2 A continuous field (a), mapped to two nominal maps (b,c) with different land use category orders.

actively handle the orders of land use orders categories in the optimization. Every individual gets assigned a land use order element that contains randomly shuffled land use category IDs (Fig. 2, b and c). The obtained integer values are re-mapped with each individual's land use order (Fig. 2). With this approach, the same function (Fig. 1, b) results in the same continuous field (Fig. 2, a), but the nominal values differ. Without re-mapping, land use with id 1 would always have a higher likelihood of neighbouring to land use 2 than to land use 5. An association of different land use orders to individuals within the population leaves the potential to find an optimal combination of land use orders and functions in the optimization.

GP procedure for the multi-objective land use allocation

The algorithm procedure starts with a random initialization of individuals until reaching the population size. This study uses the standard initialization called *ramped half and half*. It is a combination of two tree-generation algorithms *grow* and *full*, and in both the primitives and terminals are generated at random [13]. The *grow* initialization creates a sub-tree with a tree depth that is also randomly selected between a minimum and a maximum tree depth threshold. In contrast, the *full* algorithm generates a sub-tree with a depth that equals a depth threshold. Then, until a termination criterion is reached, in every generation, the algorithm evaluates the individuals with the objective function(s) and constraint(s), selects individuals for reproduction with a selection operation, generates offspring individuals in a crossover operation, and mutates the individuals in a mutation operation.

Since the algorithm is applied to multi-objective land use allocation problems, the selection operation selects individuals based on multiple objective values. We use the selection procedure from the NSGA 3 algorithm [7] that is based on the principle of Pareto efficiency and is designed to find individuals close to desired reference points. Reference points can be user-defined or distributed strategically, for example, using equal distances on the hyper-plane [5]. We refer to the original paper for a detailed description for details [7].

We use the standard GP operators *one-point crossover* and *one-point mutations* for the crossover and mutation. For example, in the *one-point crossover*, a common crossover point in the parent solutions is selected randomly, and then the corresponding sub-trees are exchanged [25]. In the *one-point mutation*, a random point of the tree is selected and then replaced with a newly generated sub-tree.

3.2 Land use allocation test problems

We use two land use allocation problems (Tab. 1) for testing the proposed method. The first test problem is a synthetic raster land use problem with 8 land use categories, two constraints, and four maximization objectives. Both problems are multi-class combinatorial

Table 1 Land use problem specifications. The raster problem is re-used; for more details see [29]. The vector problem is designed for this study.

Raster problem	Vector problem
Data and spatial representation	
Synthetic raster data serves as an initial land use map. The problem can be approached with single raster cell representation and raster patches.	Real-world parcels (vector) serve as spatial units and for the initial land use map. Sixteen land use categories associated with the parcels are mapped into seven land uses.
Land use categories	
Cropland 1-5, representing five different levels of agricultural productivity, pasture, forest, urban	Civil, rural non-forest, industrial, agriculture, forest, residential, and the last combines transport and water.
Constraints	
Land use transition constraint	*Land use transition constraint*
The transition of urban land use is restricted, forests can only be converted to pasture, and pasture cannot be converted.	Transitions of civil, and water and transport land uses are restricted; only rural-non forest be converted to forest.
Area proportion constraint	*Area proportion constraint*
Permitted ranges of 10-25% for forest, 10-30% for pasture. No area proportion constraint for other land uses.	Permitted ranges of 0-50% for industrial, 10-80% for agriculture, 15-100% for forest, 10-100% for residential. No area proportion constraint for other land uses.
Objective functions	
Max. species richness (SR)	*Max. urban compactness* (UC)
An empiric value that changes with the total forest area (unitless)	Count of adjacencies between land units of the categories civil, residential, and industrial.
Max. habitat heterogeneity (HH)	*Max. agriculture within water range* (AW)
Sum over edges between different land use types, where low-intensity land uses get higher weights than high-intensity land uses (unitless)	Area in *ha* of land use agriculture that intersects with 500-meter buffers around waters.
Max. water yield (WY)	*Max. Contiguous agriculture size* (AS)
Relative differences in evapotranspiration rates between land use types (unitless)	Average patch size in *ha* of contiguous agriculture.
Max. crop yield (CY)	*Max. distance residential to wind plants* (DRW)
Sum of all logarithmic products of cropland intensity and soil fertility over all cells (unitless).	Average distance of residential areas to the closest wind plant point in *km*.

problems; the decision variables are elements in a sequence with a length that equals the number of land units. Each element is associated with one land use category, represented as an integer value. For more specifications about the problem background and formulation, we refer to Tab. 1 and [29]. Both problems have initial land use maps. The problem instance classes and the initial land use maps are available online [2].

The second test problem is a vector land use problem with 7 land use categories, two constraints and four maximization objectives [30]. We constructed the problem for testing the algorithm's performance with parcels located in Germany.

[2] https://data.mendeley.com/datasets/4tw223jvjv

Furthermore, we generate the single objective optimal land use configurations per objective. For example, we allocated only land use *Cropland 5* while not violating the constraints to generate the optimal solution for the objective *Crop Yield*, and only the land uses *Civil*, *Industrial* and *Residential* for the objective *Urban Compactness*. The only exception is the objective *Habitat Heterogeneity*, for which we approximate the single objective optimal solution. These single objective optima are the extreme ends of the Pareto fronts. Therefore, they are insufficient to determine whether an algorithm finds the true Pareto front. However, it serves as an indicator to determine whether or not an algorithm can find optimal solutions or how far it is off from the known optima.

3.3 Design of simulation experiments and software availability

Table 2 Simulation experiment with a) Run time analysis over 10 problem sizes. b) Single-objective best solutions found by the algorithms and the known optima.

a) Run time analysis				
Problem type	Problem size	Algorithm	Nr. of generations	Pop. size
Raster	100 - 22500	GP	10	40
Raster	100 - 22500	NSGA 2 with repair mutation	10	40
Raster	100 - 22500	NSGA 2 no repair mutation	10	40
Vector	2075 - 13687	GP	10	40
Vector	2075 - 13687	NSGA 3	10	40
b) Single-objective solution comparison				
Problem type	Problem size	Algorithm	Nr. of generations	Pop. size
Raster	100	GP	100	200
Raster	100	NSGA 2 with repair mutation	100	200
Raster	10,000	GP	100	200
Raster	1,000,000	GP	100	200
Vector	13687	GP	100	200
Vector	13687	NSGA3	100	200

In the first experiment, we perform a benchmark between GP and the most commonly multi-objective land use allocation algorithm NSGA 2 on the multi-objective raster land use problem (Tab. 3). The NSGA 2 can not be applied without adaptations for solving land use allocation problems. Therefore, we compare GP to a land use allocation algorithm that bases on NSGA 2 on the multi-objective raster land use problem defined in [29]. The authors suggest a multi-objective land use allocation algorithm (CoMOLA) to solve the land use problem. The algorithm offers the option to use a repair mutation operation for patches. The spatially explicit repair functions can improve the search for optimal solutions by repairing infeasible individuals. We perform a run time benchmark on 10 problem sizes with raster dimensions from 10*10 to 150*150 cells with a step size of 10. Then, to indicate the algorithm performance of finding optimal solutions, we compare the best solutions of the single objectives from both algorithms on the 10*10 problem size to the known single objective optima.

In the second simulation experiment, we test the algorithm performance on the multi-objective vector land use problem with features representing land units. We select the NSGA 3 algorithm, the successor of NSGA 2, as benchmark algorithm for two reasons. First, we use the same selection procedure [7]. Second, NSGA 3 has proven its ability to find better-distributed solutions in Pareto fronts and has been successfully applied to solving land use allocation problems [18]. We perform a run time benchmark on 10 problem sizes ranging from 2075 to 13687 land units and compare the single objective optimal solutions for 2075 and for 13687 land units to the known single objective optima. The software used is open source and the results are fully reproducible.

The code, input data, and results files are available at Mendeley Data[2].

4 Results

4.1 Raster land use allocation problem

The scaling potential of the run times is promising. While CoMOLA with the patch repair mutation took 171 minutes to evaluate 200 individuals in 100 generations, GP needed 5 seconds for the same number of evaluations. The larger the problem instances, the larger the difference between the run times. While the run times of the CoMOLA based on the NSGA 2 algorithm increase exponentially with increasing raster problem sizes, GP run times increase linearly (Fig. 3, a).

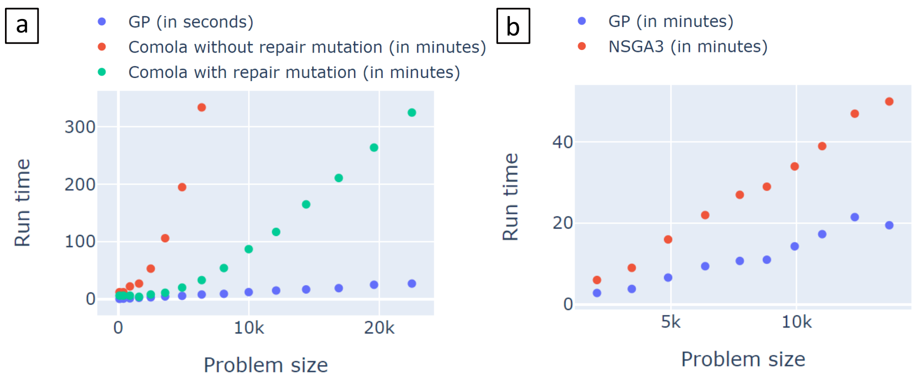

Figure 3 Total run times of with increasing land use problem sizes. a) Raster land use problem ranging from 100 (10x10) to 22500 (150x150). b) Vector land use problem with problem sizes from 2075 to 13687 land units.

The highest difference, therefore, was observed on the largest raster problem instance with 22500 (150 * 150) grid cells: here, NSGA 2 required 325 minutes, whereas GP required 27 seconds, which is 722 times faster. When applying the repair mutation on CoMOLA, the difference is even higher. The spatially explicit repair function lead to computation times that exceeded 5 hours at a problem size with 80*80 cells. In comparison: When testing GP on the problem with 1000*1000 cells leading to one million decision variables, the algorithm took 742 minutes.

The single-objective optimal solutions derived with GP (Fig. 4 and Tab. 3) prove that the algorithm can and does find global optima and solutions close to the global optimum. For obtaining the optimal solution for objective *Crop yield*, only one pixel (Fig. 4 a, top left corner) is off, where cropland 4 is allocated instead of cropland 5. All other non-constrained land uses are set to the optimal land use cropland 5. The same applies to the finer resolution of 100 * 100 pixels, where 15 out of 10000 pixels are not set to the optimal land use (Fig. 4b,

Table 3 Single objective extreme values obtained with the algorithms with the known global optima.

Raster	Size	CY [-]	HH [-]	SR [-]	WY [-]
NSGA 2 with repair mutation	10*10	125.7	84.2	9.51	97.7
GP	10*10	134.7	282.2	9.51	98.0
Known optimum	10*10	138.2	354	9.51	98.9
NSGA 2 with repair mutation	100*100	-	-	-	-
GP	100*100	13,574	24,903	23.9	9,890
Known optimum	100*100	13,615	34,778	23.9	9891
NSGA 2 with repair mutation	1000*1000	-	-	-	-
GP	1,000*1,000	1,359,403	2,703,280	60.0	989,107
Known optimum	1,000*1,000	1,359,404	3,471,380	60.0	989,108

Vector	Size	UC [-]	AS [ha]	DRW [km]	AW [ha]
NSGA3	2075	313	755	0.021	725
GP	2075	347	738	0.023	770
Known optimum	2075	372	1144	0.027	982
NSGA3	13,687	1,659	7,074	0.894	2,787
GP	13,687	1907	6,257	1.135	2,731
Known optimum	13,687	2,211	9,321	1.23	3,830

top left corner). The global optimum was obtained on both spatial resolutions for objective *Water yield* with Cropland 1 being the best land use. For objective *Species Richness*, the global optimum is obtained, but this is comparatively easy to obtain by reaching 25% of land use forest since it corresponds to the upper area constraint for land use forest. More remarkable is the produced cluster in optimal solutions for objective *Habitat heterogeneity*. For this objective, the perfect land use pattern is produced when the number of neighbours between the constrained land use forest, pasture and cropland use 1 is maximized, followed by neighbours to cropland 2 etc. GP found this pattern (Tab. 3) that seems impossible to find by CoMOLA: The best objective value obtained with GP is 3.35 times higher than the best objective value obtained by CoMOLA.

Moreover, GP is not negatively affected by larger problem instances; the convergence to the single objective optima is even better on the larger problem with 100*100 cells compared to the small problem with 10*10 cells. Even on the largest problem with 1000*1000 cells, GP found one single objective global optima and two solutions that deviate 0.001% and 0.00001% from the global optima (Tab. 3). This observation indicates the scaling potential of the algorithm's performance on a finer spatial resolution.

The single objective optima show that GP can find optimal spatial patterns for objective functions based on adjacency and connectivity for small and large land use allocation problem instances.

4.2 Vector land use allocation problem

The optimization of the vector problem requires more computation time compared to the raster problem. The computationally more expensive fitness evaluations, in which intersections, distance, and adjacency operations on features are used, are the reason for the longer run times. However, the field-evolving GP is considerably faster than the NSGA 3, and the difference increases with more land units (Fig. 3, b). On average, GP is 138% faster regardless

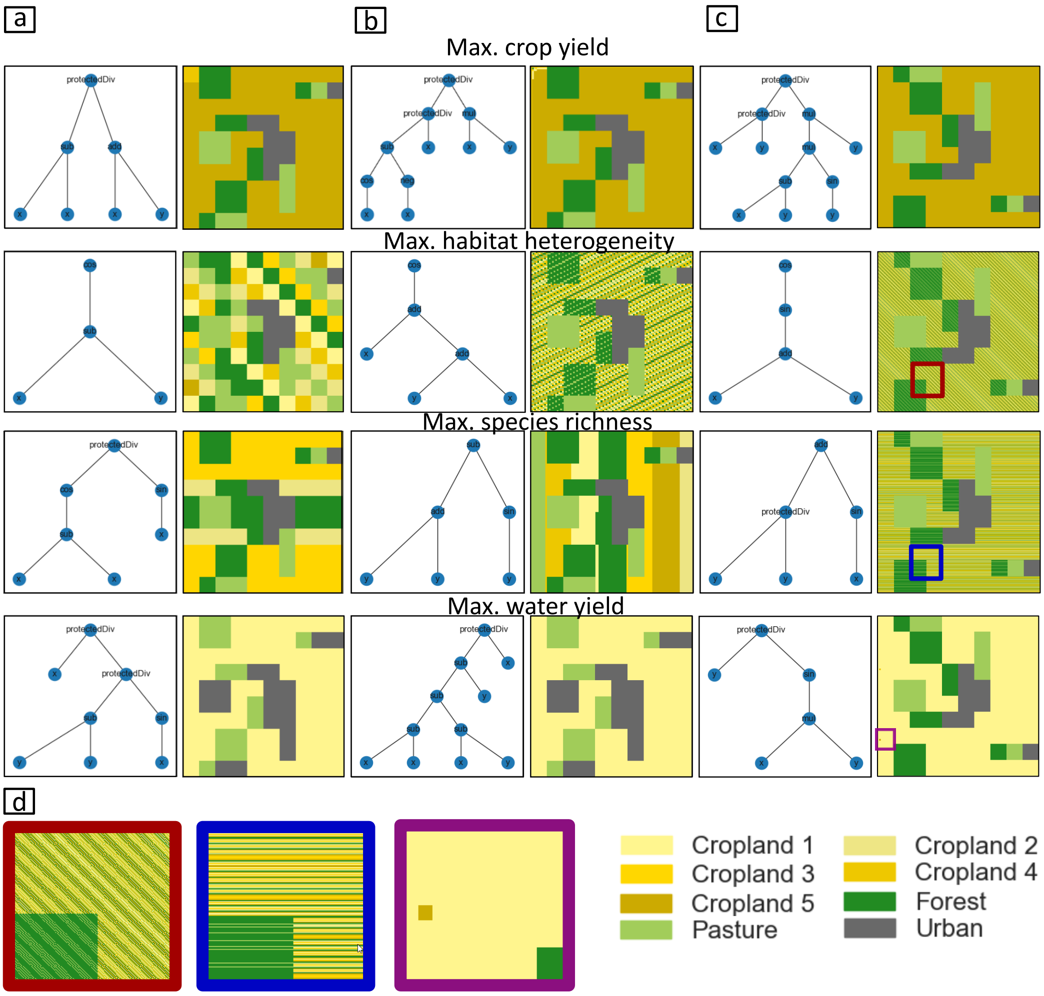

Figure 4 Single objective optimal solutions for raster land use problems with problem sizes 10x10 cells (a), 100x100 cells (b), and 1000*1000 cells (c). Close-ups (d) show produced patterns from selected regions of the 1000*1000 cell maps: The red frame shows the close-up for objective Max. habitat heterogeneity, the blue frame shows the close-up for objective Max. species richness and the purple frame shows the close-up for the objective Max. water yield.

of the problem size. However, the run time of the GP also scales well on larger problems. On the larger problem size with 13687 land units, GP took, on average, 61% less computation time per land unit than on the problem with 2075 land units.

GP did not find the global optima for the objectives in the vector problem (Tab. 3). However, GP also outperforms the NSGA 3 algorithm on the land use problem with 2075 land units (Tab. 3). The single objective optimal values of *Urban Compactness* (UC), and *Agriculture in water range* (AW) are 9.2%, 9.5%, and 6.2% better. NSGA 3 found a 2.3% better single objective optima for the *Contiguous agriculture size*. The number of optimal solutions is also higher, with 57 compared to just 7 obtained with the NSGA 3. Furthermore, GP found solutions (Fig. 5) that show spatial patterns, such as contiguous agriculture land uses, or the seemingly ordered land uses along horizontal (Fig. 5 b, last row) and the vertical axis (Fig. 5 a, last row, and b, third row)[3].

[3] Additional results, including Pareto frontiers, are available with a DOI at figshare.

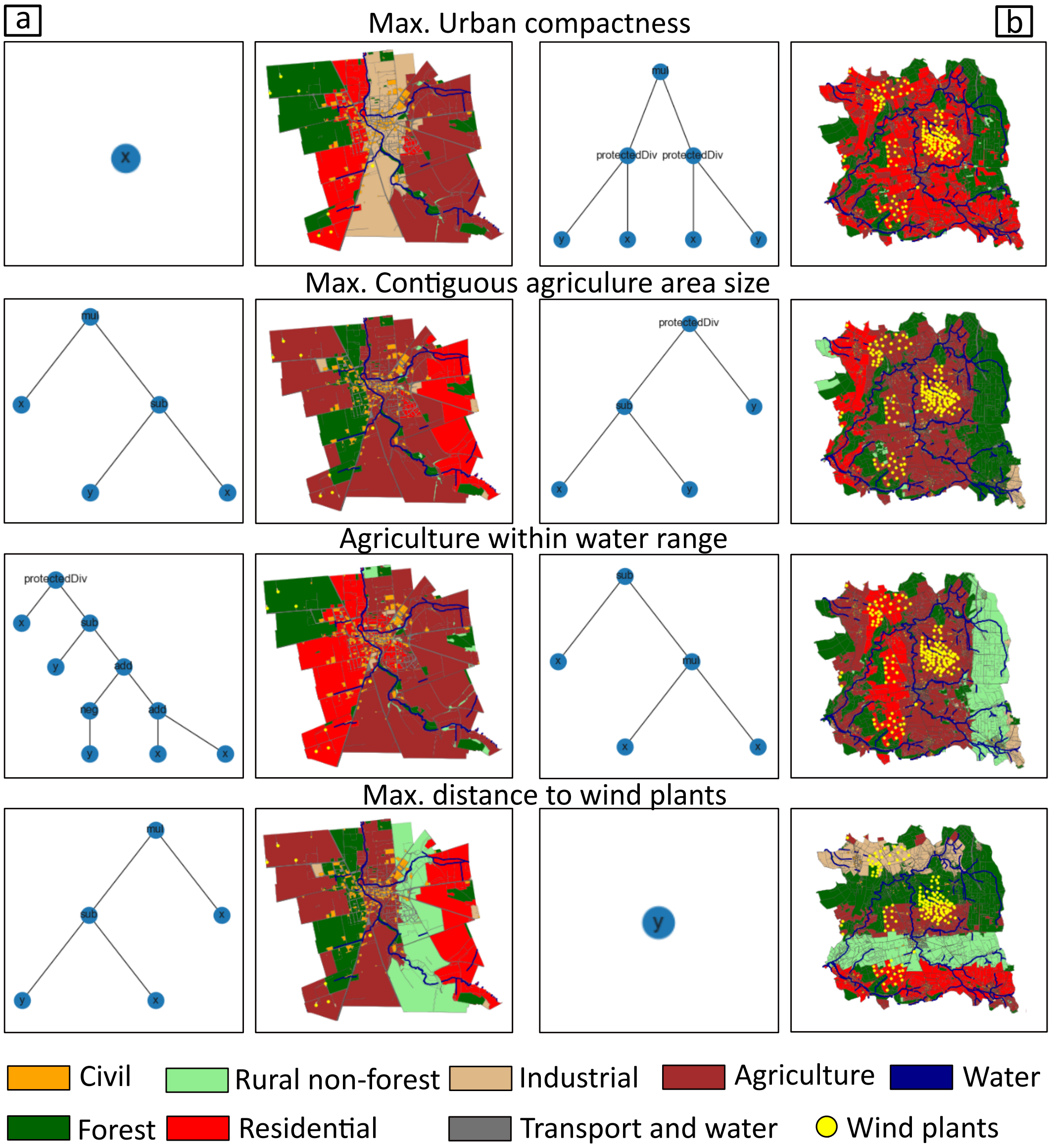

Figure 5 Single objective optimal solutions for vector land use problems with two problem sizes 2075 parcels (run time: 246 minutes) and 13687 parcels (run time: 761 minutes).

5 Discussion

5.1 Potential of encoding spatial objects as functions

The results of this study show that optimizing functions that generate continuous fields can lead to more optimal land use configurations in shorter computation times compared to using algorithms with standard encoding. The optimal land use maps produced with functions in the GP algorithms are closer to the global optima, and in many cases, GP even found the global optima. The observed scaling shows the potential for high-resolution land units and/or larger study areas, which is promising for other land use allocation problems than the ones shown here. Another example is uncertainty analysis of land use allocation optimizations, which require many optimization executions and benefit even more from the decreased computational cost [8]. Other spatial optimization problems might also be solved with the GP algorithm, e.g. 3D routing optimizations for which a sequence of 3D points is optimized instead of evolving functions [9], or facility location planning [3]. Possibly, GP can be applied to solve spatial problems that change over time by including a time dimension variable as part of the functions.

5.2 Limitations and future work

In this work, we used functions that include spatial dependencies in both x and y directions in the encoding of solutions, while a sequence of elements that represent spatial units does not. This yields a great advantage for spatial optimization problems that handle spatial objects and offers much potential for future investigation. This approach comes with disadvantages, too, e.g. the necessity to attach a random land use order to every solution to mitigate the effect of translating a continuous to a nominal variable. Investigating the random land use order association with individuals in more detail is, therefore, important for future research. For example, in our results, the portion of unique land use orders decreases over the generation and stabilizes at 40 after 50 generations. Finding out whether the observed behaviour is an anomaly or whether some land use orders are particularly suitable for solving the problem may yield important insights.

In this study, we used standard GP initialization, crossover and mutation operators and no hyper-parameter tuning to prove the general applicability of the GP algorithm on land use allocation problems. Many different initializations of trees, mutations, or crossover exist for which many parameter settings are possible, and some operators and parameter settings may yield better results for land use allocation problems or other spatial optimization problems. One parameter that should be tuned is the maximum tree depth. This parameter was set to 8, but the maximum tree depth in the optimal solutions was 5. In the vector problem, the tree depth was even shallower; some only had one terminal (Fig. 5). The parameter tuning is, therefore, future work to further improve the algorithm performance.

Lastly, the better performance on the raster problem compared to the vector problem leaves room for further analysis. The static boundaries of the features might be the reason for this observation: While GP could generate patterns and clusters that potentially benefit objectives in the raster case, that positive characteristic of the algorithm can not be realized in the vector case where the object extents are set. Another reason may be the usage of polygon centroids as x and y inputs to the function. A different mapping is possibly better for considering the whole feature's extent, e.g., using multiple points per polygon as input.

6 Conclusion

Standard land use allocation optimization algorithms cannot cope with large land use allocation problems due to the solution encoding. Using function as solution encoding proved to solve land use problems more efficiently. The functions represent spatial fields that are mapped to nominal land use maps. We solve the identified mapping problem from continuous fields to nominal maps by associating random land use orders with the individuals of the GP population.

GP proved its ability to alleviate exponentially increasing run times of the standard encoding scheme on a raster and a vector problem. While the computation time using the standard solution encoding increased exponentially, the computation time using GP increased linearly. As a consequence, the reduction of computation time increases exponentially with larger problem instances, too. On the largest raster problem instance, GP was up to 722 times faster than the NSGA 2 land use allocation algorithm. The difference in computation time further increases when comparing GP to the standard encoding coupled with spatially explicit operators.

Moreover, the improvement in computation time does not affect the algorithm's performance in finding better solutions than the benchmark algorithms. GP obtained better single-objective solutions than NSGA 2 and NSGA 3 on six out of eight objectives of the two benchmark problems. Moreover, GP found the global single objective optima for three objectives of the raster problem with 10*10 cells and 100*100 cells. Even on the 1000*1000 single-cell raster problem, one global optimum was found and two near-optimal (deviation of 0.001% and 0.00001% from global optima). The highest increased performance was obtained for the objective *Habitat Heterogeneity* of the raster problem that requires finding a highly complex spatial pattern of adjacent land uses. Also, GP found contiguous clusters required to find optimal solutions to four other objectives. This shows that GP can produce land use maps with spatial patterns that involve adjacency and connectivity.

We conclude that evolving functions enable more efficient land use allocation optimizations in the future and that the approach is a promising method for other spatial optimization problems.

References

1. Kai Cao, Bo Huang, Shaowen Wang, and Hui Lin. Sustainable land use optimization using boundary-based fast genetic algorithm. *Computers, Environment and Urban Systems*, 36(3):257–269, 2012. doi:10.1016/j.compenvurbsys.2011.08.001.
2. Kai Cao, Muyang Liu, Shu Wang, Mengqi Liu, Wenting Zhang, Qiang Meng, and Bo Huang. Spatial multi-objective land use optimization toward livability based on boundary-based genetic algorithm: A case study in singapore. *ISPRS International Journal of Geo-Information*, 9(1):40, 2020. doi:10.3390/ijgi9010040.
3. Richard L. Church and Alan T. Murray. *Business Site Selection, Location Analysis and GIS*. John Wiley & Sons, Inc, Hoboken, NJ, USA, 2008. doi:10.1002/9780470432761.
4. Felix Creutzig, Christopher Bren d'Amour, Ulf Weddige, Sabine Fuss, Tim Beringer, Anne Gläser, Matthias Kalkuhl, Jan Christoph Steckel, Alexander Radebach, and Ottmar Edenhofer. Assessing human and environmental pressures of global land-use change 2000–2010. *Global Sustainability*, 2, 2019. doi:10.1017/sus.2018.15.
5. Indraneel Das and John E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998. doi:10.1137/S1052623496307510.
6. Kalyanmoay Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi:10.1109/4235.996017.
7. Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014. doi:10.1109/TEVC.2013.2281535.
8. Moritz Hildemann and Judith A. Verstegen. Quantifying uncertainty in pareto fronts arising from spatial data. *Environmental Modelling & Software*, 141:105069, 2021. doi:10.1016/j.envsoft.2021.105069.
9. Moritz Hildemann and Judith A. Verstegen. 3d-flight route optimization for air-taxis in urban areas with evolutionary algorithms and gis. *Journal of Air Transport Management*, 107:102356, 2023. doi:10.1016/j.jairtraman.2022.102356.
10. John H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* u Michigan Press, 1975.

11 Vijay Ingalalli, Sara Silva, Mauro Castelli, and Leonardo Vanneschi. A multi-dimensional genetic programming approach for multi-class classification problems. In Miguel Nicolau, Krzysztof Krawiec, Malcolm I. Heywood, Mauro Castelli, Pablo García-Sánchez, Juan J. Merelo, Victor M. Rivas Santos, and Kevin Sim, editors, *Genetic Programming*, volume 8599 of *Lecture Notes in Computer Science*, pages 48–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. doi:10.1007/978-3-662-44303-3_5.

12 Konstantin Klemm, Anita Mehta, and Peter F. Stadler. Landscape encodings enhance optimization. *PloS one*, 7(4):e34780, 2012. doi:10.1371/journal.pone.0034780.

13 John R. Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2), 1994. doi:10.1007/BF00175355.

14 Werner Kuhn. Core concepts of spatial information for transdisciplinary research. *International Journal of Geographical Information Science*, 26(12):2267–2276, 2012. doi:10.1080/13658816.2012.722637.

15 Arika Ligmann-Zielinska, Richard L. Church, and Piotr Jankowski. Spatial optimization as a generative technique for sustainable multiobjective land–use allocation. *International Journal of Geographical Information Science*, 22(6):601–622, 2008. doi:10.1080/13658810701587495.

16 Hongjiang Liu, Fengying Yan, and Hua Tian. Towards low-carbon cities: Patch-based multi-objective optimization of land use allocation using an improved non-dominated sorting genetic algorithm-ii. *Ecological Indicators*, 134:108455, 2022. doi:10.1016/j.ecolind.2021.108455.

17 Yaolin Liu, Jinjin Peng, Limin Jiao, and Yanfang Liu. Psola: A heuristic land-use allocation model using patch-level operations and knowledge-informed rules. *PloS one*, 11(6):e0157728, 2016. doi:10.1371/journal.pone.0157728.

18 Jamshid Maleki, Zohreh Masoumi, Farshad Hakimpour, and Carlos A. Coello Coello. Many–objective land use planning using a hypercube–based nsga–iii algorithm. *Transactions in GIS*, 26(2):609–644, 2022. doi:10.1111/tgis.12876.

19 Shahrul Badariah Mat Sah, Vic Ciesielski, Daryl D'Souza, and Marsha Berry. Comparison between genetic algorithm and genetic programming performance for photomosaic generation. In Xiaodong Li, Michael Kirley, Mengjie Zhang, David Green, Vic Ciesielski, Hussein Abbass, Zbigniew Michalewicz, Tim Hendtlass, Kalyanmoy Deb, Kay Chen Tan, Jürgen Branke, and Yuhui Shi, editors, *Simulated Evolution and Learning*, volume 5361 of *Lecture Notes in Computer Science*, pages 259–268. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi:10.1007/978-3-540-89694-4_27.

20 James McDermott, David R. White, Sean Luke, Luca Manzoni, Mauro Castelli, Leonardo Vanneschi, Wojciech Jaskowski, Krzysztof Krawiec, Robin Harper, Kenneth de Jong, and Una-May O'Reilly. Genetic programming needs better benchmarks. In Terence Soule and Jason H. Moore, editors, *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 791–798, New York, NY, USA, 07072012. ACM. doi:10.1145/2330163.2330273.

21 Ibrahim H. Osman and James P. Kelly, editors. *Meta-Heuristics*. Springer US, Boston, MA, 1996. doi:10.1007/978-1-4613-1361-8.

22 Tingting Pan, Yu Zhang, Fenzhen Su, Vincent Lyne, Fei Cheng, and Han Xiao. Practical efficient regional land-use planning using constrained multi-objective genetic algorithm optimization. *ISPRS International Journal of Geo-Information*, 10(2):100, 2021. doi:10.3390/ijgi10020100.

23 Edzer J. Pebesma. The role of external variables and gis databases in geostatistical analysis. *Transactions in GIS*, 10(4):615–632, 2006. doi:10.1111/j.1467-9671.2006.01015.x.

24 Fernando Peres and Mauro Castelli. Combinatorial optimization problems and metaheuristics: Review, challenges, design, and development. *Applied Sciences*, 11(14):6449, 2021. doi:10.3390/app11146449.

25 Riccardo Poli and W. B. Langdon. Genetic programming with one-point crossover. In P. K. Chawdhry, R. Roy, and R. K. Pant, editors, *Soft Computing in Engineering Design and Manufacturing*, pages 180–189. Springer London, London, 1998. doi:10.1007/978-1-4471-0427-8_20.

26 Mostafizur Rahman and György Szabó. Multi-objective urban land use optimization using spatial data: A systematic review. *Sustainable Cities and Society*, 74:103214, 2021. doi:10.1016/j.scs.2021.103214.

27 Franz Rothlauf. Optimization methods. In Franz Rothlauf, editor, *Design of Modern Heuristics*, Natural Computing Series, pages 45–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi:10.1007/978-3-540-72962-4_3.

28 Mingjie Song and DongMei Chen. A comparison of three heuristic optimization algorithms for solving the multi-objective land allocation (mola) problem. *Annals of GIS*, 24(1):19–31, 2018. doi:10.1080/19475683.2018.1424736.

29 Michael Strauch, Anna F. Cord, Carola Pätzold, Sven Lautenbach, Andrea Kaim, Christian Schweitzer, Ralf Seppelt, and Martin Volk. Constraints in multi-objective optimization of land use allocation – repair or penalize? *Environmental Modelling & Software*, 118:241–251, 2019. doi:10.1016/j.envsoft.2019.05.003.

30 Daoqin Tong and Alan T. Murray. Spatial optimization in geography. *Annals of the Association of American Geographers*, 102(6):1290–1309, 2012. doi:10.1080/00045608.2012.685044.