Strauch, M., **Cord, A.F.**, **Pätzold, C.**, **Lautenbach, S.**, **Kaim, A.**, **Schweitzer, C.**, **Seppelt, R.**, **Volk, M.**
(2019):
Constraints in multi-objective optimization of land use allocation – repair or penalize?
Environ. Modell. Softw. **118** , 241 – 251

http://dx.doi.org/10.1016/J.ENVSOFT.2019.05.003

Constraints in multi-objective optimization of land use allocation – repair or penalize?

Authors: Michael Strauch[1], Anna F. Cord[1], Carola Pätzold[1], Sven Lautenbach[1,2,3], Andrea Kaim[1], Christian Schweitzer[1,4], Ralf Seppelt[1,5], Martin Volk[1,5]

[1] UFZ - Helmholtz-Centre for Environmental Research, Department Computational Landscape Ecology, Permoserstraße 15, 04318 Leipzig (Germany)

[2] University of Bonn, Nussallee 1, 53115 Bonn (Germany)

[3] University of Heidelberg, GIScience Group, Im Neuenheimer Feld 348, 69120 Heidelberg

[4] German Environment Agency, Wörlitzer Platz 1, 06844 Dessau-Roßlau (Germany)

[5] Institute of Geoscience & Geography, Martin Luther-University Halle-Wittenberg, 06099 Halle (Saale), Germany

Corresponding author:
Michael Strauch, michael.strauch@ufz.de, tel.: +49 341 235 1464, fax: +49 341 235 1939

Abstract

Combining simulation models and multi-objective optimization can help solving complex land use allocation problems by considering multiple, often competing demands on landscapes, such as agriculture, (drinking) water provision, or biodiversity conservation. The search for optimal land use allocations has to result in feasible solutions satisfying “real-world” constraints. We here introduce a generic and readily applicable tool to integrate user-specific spatial models (e.g. assessing different ecosystem services) for a *Constrained Multi-objective Optimization of Land use Allocation* (CoMOLA). The tool can handle basic land use conversion constraints by either a newly and specifically developed method to repair infeasible solutions or by penalizing constraint violation. CoMOLA was systematically tested for different levels of complexity using a virtual landscape and simple ecosystem service and biodiversity models. Our study shows that using repair mechanisms seems to be more effective in exploring the feasible solution space while penalizing constraint violation likely results in infeasible solutions.

Keywords: Land use allocation, ecosystem services, biodiversity, multi-objective optimization, constraint-handling

Software availability:

Name of Software: CoMOLA
Programming language: Python
Packages used: inspyred
Developers: Carola Pätzold, Christian Schweitzer, Sven Lautenbach, Michael Strauch
Contact address: Helmholtz Centre for Environmental Research – UFZ, Department of Computational Landscape Ecology, Permoserstrasse 15, D-04318 Leipzig, Germany. E-mail: michael.strauch@ufz.de
Availability: https://github.com/michstrauch/CoMOLA

1. Introduction

Achieving the goals of global environmental sustainability requires the conservation of biodiversity as well as ensuring the provision of ecosystem services, such as drinking water, food, or timber. While these objectives are inextricably linked, they are often studied and managed separately from each other (Liu et al., 2015). Sustainable resource use and management strategies that enhance landscape multifunctionality require not only understanding the manifold interactions among multiple demands but also finding solutions to minimize their trade-offs. While progress has been made in the model-based quantification of land use effects on ecosystem services and biodiversity over the past few years, the question of "[w]here to put things?" (Polasky et al., 2008) is still an important challenge.

Land allocation problems can be solved by model-based approaches combined with either scenario analysis (e.g. Fontana et al., 2013), backcasting approaches (e.g. Brunner et al., 2016), or multi-objective optimization techniques. Optimization methods are capable to explore a large number of land use/land management configurations (Memmah et al., 2015; Seppelt et al., 2013). Combined with simulation models (e.g. hydrological, biodiversity or socio-economic models), they are applicable for a wide variety of problems. Existing studies on land use optimization were reviewed, for example, by Kaim et al. (2018), Kanter et al. (2016) and Memmah et al. (2015). Optimization applications differ in terms of scale (from single fields/farms to large river basins), the objectives considered (e.g. agricultural productivity, water quality, biodiversity, recreational value) and the timing of preference articulation of stakeholders and decision makers (before, during or after the optimization process). Such multi-objective land use allocation problems can be solved either by integrating all objectives into one single function following a weighted sum approach (Marler and Arora, 2010) or by Pareto-based methods where all objectives are treated individually using the concept of Pareto-optimality (Deb, 2014). The latter approach provides a set of 'Pareto-optimal' solutions, i.e. solutions for which no objective can be further improved without compromising at least one of the other objectives. As those solutions are usually only approximations of true Pareto-optimality, we refer to them as non-dominated solutions. Pareto-based optimization methods have been used in several land use optimization studies, e.g. in Bennett et al. (2004), Cao et al. (2011), Chikumbo et al. (2015), Duh and Brown (2007), Groot et al. (2012), Huang et al. (2013), Lautenbach et al. (2013), Roberts et al. (2011) and Schwaab et al. (2017, 2018a). The advantage of non-dominated solutions is that they directly illustrate the best possible trade-offs among conflicting objectives. From such a set of best alternatives, decision makers can discuss and select appropriate solutions according to their preferences (Cord et al., 2017). To obtain non-dominated solutions, population-based meta-heuristics, i.e. methods based on evolutionary computation or swarm intelligence, are particularly suitable because they generate and evaluate a set of potential solutions in parallel and approach the Pareto front within a single run of the algorithm (Boussaïd et al., 2013; Coello

Coello et al., 2007). Identifying non-dominated solutions also allows evaluating certain projections on future land use change regarding their overall optimality (Seppelt et al., 2013; Verstegen et al., 2017).

Such non-dominated solutions, however, have little value for decision making if they ignore real-world constraints. For example, depending on rules and regulations in the study region, grassland may be transformed into forest but not into cropland or urban areas. In the same case study context, no specific transition rules may exist for other land use classes (e.g. cropland). Alternatively, the goal might be to efficiently allocate a specified demand for urban development in a district or municipality (Schwaab et al., 2017). This problem is different from the problem of constraining based on objective values such as setting threshold values for one or several objectives. In contrast to constraining the objective values (e.g. a minimum level of species richness or a maximum value for nitrate concentration in the groundwater), constraints are put on the control variables here. For land use allocation problems, these constraints on the control variables could involve the minimum and maximum allowed total area of certain land use classes or transition rules controlling which classes are allowed to be replaced by others and which not. Since evolutionary algorithms are by nature unconstrained, it is necessary to find proper ways of incorporating real-world constraints (Coello Coello, 2002). There are three constraint-handling methods commonly applied in land use optimization studies:

(1) penalty functions that degrade the fitness value of an infeasible solution (e.g. Chikumbo et al., 2015; Shaygan et al., 2014; Stewart et al., 2004),
(2) feasibility operators which create feasible-only child solutions (e.g. García et al., 2017; Karakostas and Economou, 2014) and
(3) repair mechanisms for infeasible individuals (e.g. Cao et al., 2012; Datta et al., 2007; Schwaab et al., 2018b).

The latter two approaches require modifications of the genetic algorithm which are often specific for the optimization problem at hand (Schwaab et al., 2018b). Their adaptation to different objective functions, different land use types including their transition or total area rules may require extensive effort and programming skills - if the source code is available at all.

To foster the use of population-based optimization in landscape management, we here present a generic tool for *Constrained Multi-objective Optimization of Land use Allocation* (CoMOLA), including constraint-handling for the control variables with large flexibility and broad applicability. CoMOLA employs the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) developed by Deb et al. (2002). Among multi-objective evolutionary algorithms NSGA-II is the most often used procedure for solving spatial allocation problems (Malczewski and Rinner, 2015). We enhanced the algorithm by implementing two types of constraint-handling methods, (1) a novel repair algorithm specifically developed for land use transition and composition constraints and (2) constrained tournament selection, a commonly used penalty function approach (Deb, 2000; Osyczka and Krenich, 2000).

The objective of this research paper is to introduce the functionality of CoMOLA and compare the performance of CoMOLA's repair algorithm with the established method of constraint tournament selection. To test all important facets of the tool, we used a virtual land use map which was designed to optimize multiple conflicting objectives (maximizing crop and water yield as well as two biodiversity indicators) while considering basic land conversion constraints (total area and transition rules). Our optimization experiments include different levels of complexity with respect to the number of objectives, the strength of the pre-defined constraints and the number of spatial units to be optimized.

2. Description of CoMOLA

CoMOLA is a free Python tool to optimize raster maps for multiple objectives. It is based on the open source 'inspyred' Python library (Garrett, 2012), which is extended to include functions for reading, encoding, and writing raster maps, as well as algorithms to consider constraints during the optimization procedure. It is platform independent and allows for the integration of any model which reads gridded maps as input data, for instance for estimating ecosystem service indicators. CoMOLA can be used immediately by inputting a raster map representing the baseline situation (e.g. for land use), ready-to-run models written in R (R Core Team, 2016) or Python, and (optional) information on constraints.

As constraints for the control variables, the tool is able to consider both (1) transition rules defining the possible land use transformations and (2) minimum and maximum area proportions for each land use class within the study area. All relevant settings, such as paths to input data and models as well as optimization-specific parameters (e.g. population size, crossover and mutation rates) and settings related to constraint-handling and raster map-analysis are managed in a single control file. As the framework allows users to integrate any kind of simulation models (e.g. process-based, statistical, or both at the same time), it can be a useful tool for a broad range of spatial allocation studies. For more details and examples how to adapt and apply CoMOLA the reader is referred to the user guide on GitHub (https://github.com/michstrauch/CoMOLA). The optimization procedure is summarized in Figure 1a and can be briefly described as follows (assuming a land use allocation problem):

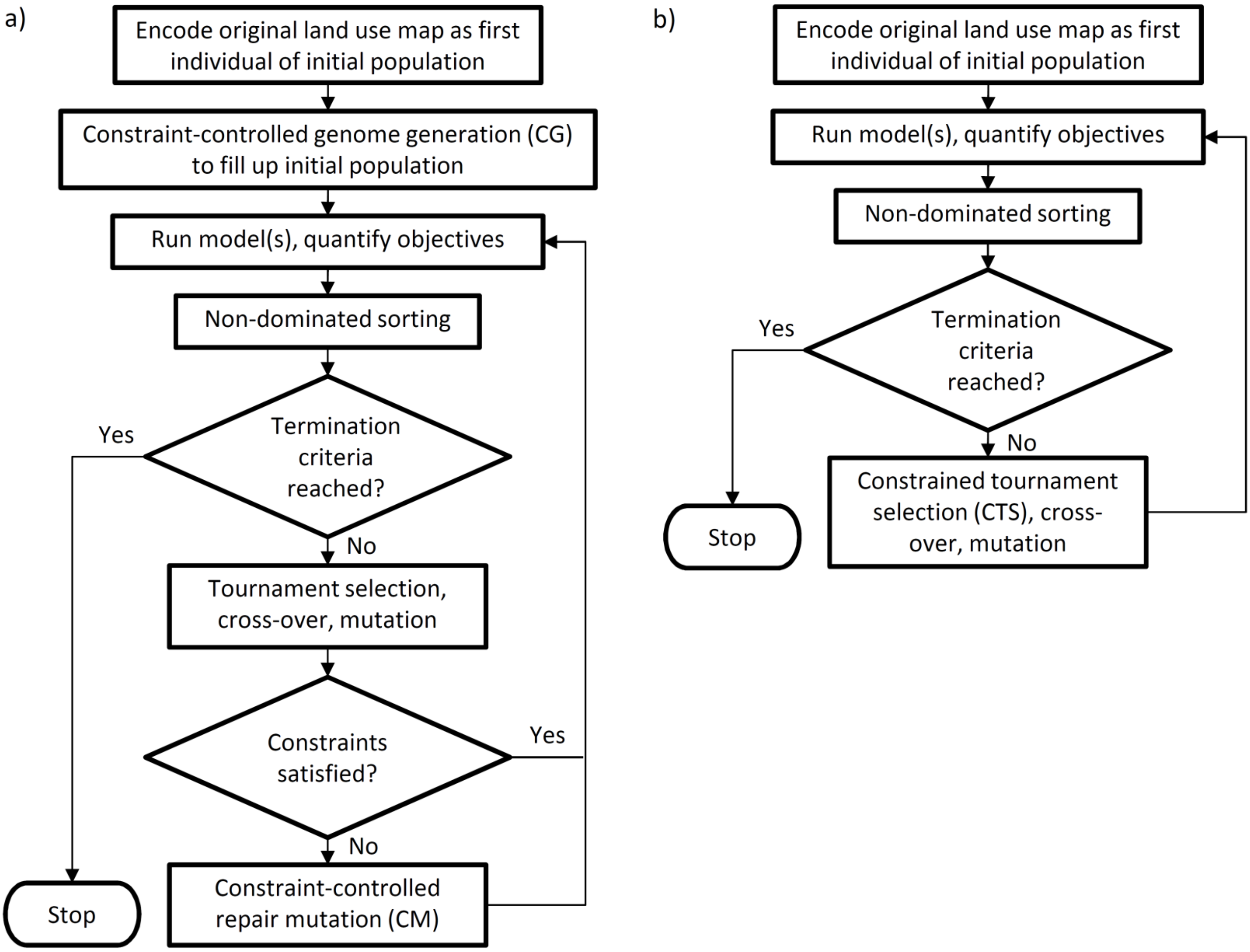

Figure 1: Flowchart of CoMOLA a) using a combination of constraint-controlled genome generation and repair mutation (CG-CM) and b) using constrained tournament selection (CTS) as constraint handling method.

(1) Preprocessing - encoding the original raster map

To decrease the number of spatial units (and hence the computational effort), the algorithm first transforms the input raster map representing the current status of land use into a patch or cluster map (Fig. 2) where neighboring raster cells of the same type are aggregated (cf. Holzkämper and Seppelt, 2007; Liu et al., 2016). The patch ID map is then encoded as a string of integers (each value is called a gene, representing the land use of a patch) that forms the genome of the initial individual, i.e. the first individual of the initial population. However, users may also specify their own patch ID map to control the aggregation of cells into patches. The patch-ID map hence allows complete flexibility to delineate spatial units; if desired, the user can also conduct a cell-level optimization. The patch-ID map can be thought of as a polygon map that defines the units that the optimizer modifies with respect to land use. The approach combines the possibility to use irregularly shaped patches as modeling units while allowing the use of raster processing functions which are more efficient for neighborhood analysis problems.

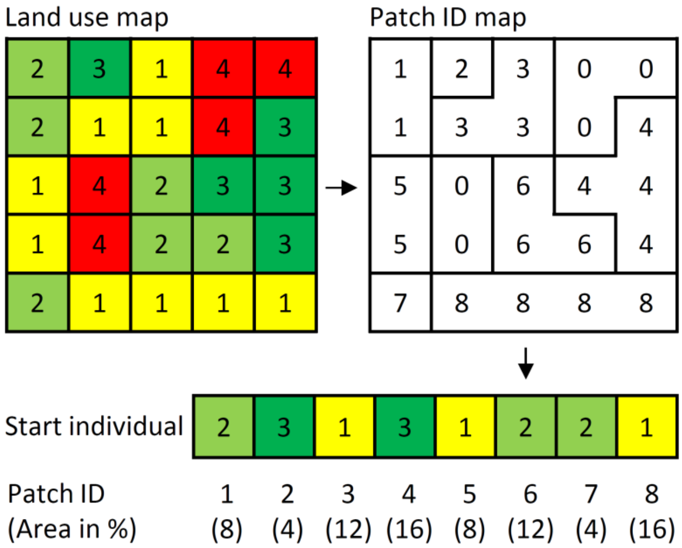

Figure 2: Example scheme for encoding the input land use map into the genome of the initial individual. The numbers and the corresponding colors in the virtual land use map indicate different land cover types or uses. In the patch ID map neighboring cells of the same land use are aggregated to a single patch with a unique patch ID. The initial genome is created by transforming the patch ID map into a string (from top left to bottom right) including information about the land use and share of the total area of each patch ID. Land use classes which cannot change at all according to pre-defined transition rules (here value 4, see Fig. 3) will be excluded from the genome.

(2) Constraint-controlled genome generation (CG) to fill up the initial population

In the second step, all further individuals of the initial population are generated - the original land use map is always included in the initial population. The (user-defined) size of the population depends on the nature of the problem, but should usually not be higher than a few hundred individuals (Malczewski and Rinner, 2015). Without considering constraints, the genome generation would be completely at random and based on a uniform sampling among all available land use classes. If constraints are defined, the initialization uses a constraint-controlled genome generation (CG) algorithm to ensure all individuals of the first population are feasible and fulfill all given constraints. It is known that the search process of an optimizer becomes more efficient if its initial guess is feasible or near-feasible (Datta et al., 2012). CG operates on a gene-by-gene (i.e., if not defined otherwise, a patch-wise) basis to generate a genome. For each gene, the random choice of a land use class is constrained according to the pre-defined transition rules (Fig. 3) and a tabu memory, which is a list of already identified infeasible gene sequences and visited feasible solutions. Each time a land use class is assigned to a gene, the algorithm examines the genome sequence generated up to that point for any violation of the total area constraint (if pre-defined) for this land use class. If there is no violation, the genome generation proceeds with the next gene (i.e. the next patch ID). If the total area constraint is violated, the gene sequence is added to the tabu memory and the algorithm returns to a suitable previous gene. The search is then repeated for a feasible land use class according to the transition rules and the updated tabu memory.

Transition matrix

	1	2	3	4
1	1	1	1	0
2	0	1	1	0
3	0	0	1	0
4	0	0	0	1

Area proportion

	1	2	3	4
Min	0	0	20	0
Max	100	100	35	100

Figure 3: Example land use change constraints. Transition rules (left) can be defined with a transition matrix where value 1 allows and 0 prohibits the transition. For example, land use type 2 can be transformed into land use types 2 and 3 but not into 1 and 4. Area proportion values (right) represent minimum and maximum percentages of the total area for each land use type.

In addition, the user can choose to include boundary solutions in the initial population, which are solutions at the boundary of the feasible solution space. In cases where total area constraints are defined, two boundary solutions are automatically generated within CoMOLA for each land use class: one solution with the minimum permissible and another with the maximum permissible coverage of that respective land use class. Boundary solutions may have a good fitness for at least one objective and good initial estimates (also referred to as seeds) may generate better solutions with faster convergence (Friedrich and Wagner, 2015).

(3) Run model(s), quantify objectives.

Fitness values for each individual (i.e., each land use map) of the population are estimated through user-defined models. Since the search ability of Pareto-dominance based evolutionary algorithms such as NSGA-II deteriorates with increasing number of objectives (Ishibuchi et al., 2008), we recommend using a maximum of four objectives. However, due to the modular architecture of CoMOLA, it is possible to implement other optimization algorithms, such as NSGA-III (Deb and Jain, 2014), to handle more than four objectives. For quantifying the single objectives (e.g. agricultural productivity, water quality, biodiversity, etc.) any type of model can be integrated as long as the model is able to handle a raster map as land use input and produces only one (aggregated) value per objective that is representative for the whole map.

(4) Non-dominated sorting, tournament selection, crossover, mutation

Based on the fitness values of each member of the current population, NSGA-II applies a non-dominated sorting scheme, archives the best solutions, and uses binary tournament selection, mating (crossover), and mutation operators to create an offspring population as described in Deb et al. (2002).

(5) Constraint-controlled repair mutation (CM)

Even if the parent population exclusively consisted of feasible individuals, the offspring population likely contains infeasible solutions due to crossover and mutation (Yoon and Kim, 2014). Infeasible individuals of the offspring population are transformed into feasible individuals using a constraint-controlled repair mutation (CM) algorithm which operates patch-wise, exactly as constraint-controlled genome generation (step 2), but with the premise that only a minimum number of genes should be modified in order to make an infeasible genome feasible. Hence, the repaired (feasible) individuals remain as similar as possible to the original (infeasible) solutions suggested by NSGA-II. CM is thus a further operator besides crossover and mutation that repairs infeasible offspring. The implemented tabu memory ensures both an efficient repair of infeasible individuals and uniqueness among all (feasible) individuals generated, avoiding redundant model evaluations and fostering diversity in the solution space.

Steps 3 to 5 are repeated until a termination criterion is reached (e.g. the user-defined maximum number of generations or any other pre-defined terminator of the ‘inspyred’ package). The combination of CG and CM (CG-CM) is a novel and straight-forward constraint-handling method, specifically designed for CoMOLA to handle transition and area proportion constraints in land use optimization studies. Moreover, CoMOLA can employ constraint tournament selection (CTS) as an alternative method (Fig. 1b). CTS has been suggested by Osyczka and Krenich (2000) and Deb (2000) and represents an enhanced version of the binary tournament selection, where two solutions are picked from the parent population and the better solution is chosen for mating. Without using repair mutation operators, each solution can be either feasible or infeasible. Feasible solutions are preferred over infeasible solutions. If both solutions are feasible, CTS prefers the one with better fitness. In case both solutions competing in a tournament are infeasible, CTS prefers the solution with smaller constraint violation (Deb et al., 2002). Such a modification of the binary tournament selection is also proposed for the recent NSGA-III algorithm to solve constrained many-objective problems (Jain and Deb, 2014). In CoMOLA, constraint violation (Δ_O) is composed of two terms, one refers to the land use area violation (Δ_A) and the other to the violation of transition rules (Δ_T):

$$\Delta_O = \Delta_A + \Delta_T \qquad \text{(Eq. 1),}$$

with

$$\Delta_A = \begin{cases} \frac{\sum_{k=1}^{K} l_k}{C}, & C > 0 \\ 0, & C = 0 \end{cases} \qquad \text{(Eq. 2),}$$

and

$$\Delta_T = \sum_{j=1}^{J} \Phi_j \quad , \Phi_j := \begin{cases} p_j, & \text{transition rule is violated for patch } j \\ 0, & \text{else} \end{cases} \qquad \text{(Eq. 3).}$$

K is the total number of land use classes with $k \in \{1, \dots, K\}$, while C is the number of land use classes violating the area constraint. l denotes the area deviation of land use k from its permissible min-max range (in % of the total study area), i.e., $l > 0$ if the area constraint for land use k is violated and $l = 0$ if not. J is the total number of land use patches with $j \in \{1, \dots, J\}$ and p_j denotes the area of patch j where transition rules are violated (in % of the total study area).

3. Optimization experiment

3.1 Input land use data and objective functions

For the performance test, a virtual landscape of 10x10 raster cells was designed (Fig. 4a), for which a discrete encoding strategy was applied defining eight land use classes (1-5: cropland ranging from lowest (1) to highest production intensity (5), 6: pasture, 7: forest, 8: urban area). We assumed a cell size within the range of 100 to 10,000 m², as is typical for landscape-scale assessments.

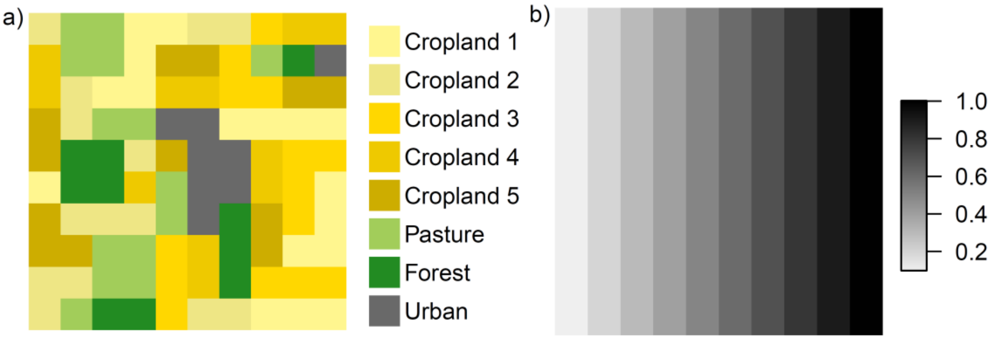

Figure 4: Input land use map (a) and soil fertility map (b), both artificially generated for the optimization experiments. Cropland is distinguished by production intensity levels, ranging from 1 (lowest level) to 5 (highest level). Hypothetical soil fertility ranges from 0 (lowest) to 1 (highest).

We considered up to four contrasting objectives including agricultural production, water yield**,** and biodiversity while taking into account land use intensity, landscape composition and landscape configuration as suggested by Seppelt et al. (2016). All four objectives are to be maximized. For the ease of evaluation and interpretation, each objective was quantified by simple yet plausible conceptual models as summarized in Table 1.

Table 1: Conceptual models for the objective functions to be maximized during optimization.

Objective, formula and theoretical min/max	**Explanation**
Crop yield (CY) $CY = \sum_{i=1}^{I} log(P_i(1 + F_i)$ Min: 0 (no cropland) Max: 203 (100% cropland with intensity level 5)	Crop yield was modeled as a function of production intensity level P (with a value of 0 for pasture, forest, and urban land use classes, and a value of 1 for cropland 1, 2 for cropland 2, etc.) and soil fertility F (ranging from 0.1 to 1, Fig. 4b), summarized over all grid cells i. The model results in hypothetical crop yields per raster cell as shown in Figure A.1 in the Supplementary.
Forest species richness (SR) $SR = cA^z$ Min: 0 (no forest) Max: 12.56 (100 % forest)	Forest species richness was a function of forest area A, based on empirical relationships between habitat area and species richness (cf. MacArthur and Wilson, 1967), where A equals the number of grid cells containing forest, c is a constant (y-intercept) and z is the slope of the species-area-relationship in log-log space. We assigned the commonly used values of 5 and 0.2 to c and z, respectively.
Habitat heterogeneity (HH) $HH = \sum_{L=1}^{5} E_L/L$ Min: 0 (only one land use class existing, regardless which one) Max: 116 (maximum heterogeneity of land use patches with lowest intensity) or 180 in case of a cell-level optimization (ideal chessboard pattern of land use classes with lowest intensity)	Habitat heterogeneity, as a universal driver of species richness (Stein et al., 2014), was estimated based on the number of edges (E) between grid cells of different land use, namely between forest and pasture, forest and cropland, as well as pasture and cropland. L represents land use intensity along edges with forest or pasture with a value of 1 for edges between forest and pasture as well as between forest or pasture and cropland 1, a value of 2 for edges between forest or pasture and cropland 2, etc., and a value of 5 for edges between forest or pasture and cropland 5. Edges with urban cells were ignored. Through dividing by L, edges of cells with higher land use intensity were given a lower weight
Water yield (WY) $WY = \sum_{c=1}^{7} A_c/K_c$ Min: 87.72 (100 % forest) Max: 111.11 (100 % cropland with intensity level 1)	Water yield was estimated based on average crop coefficients (K_c), i.e. on values reflecting relative differences in land-use specific evapotranspiration rates (Allen et al., 1998). In our case, K_c values increased in the following order for the seven land use classes considered (excluding urban): cropland 1, cropland 2, cropland 3, pasture, cropland 4, cropland 5, forest. The lower the evapotranspiration rates, the more water was assumed to be available for groundwater recharge and runoff. A is the area (number of grid cells) of land use class c.

3.2 Experimental design

We compared our novel repair algorithm (CG-CM) with CTS, a standard approach to handle constraints, and a combination of both approaches where CTS is based on a feasible start population (CG-CTS). The comparison considered different levels of complexity, as briefly explained in the following and illustrated in Figure 5.

Individual size (number of spatial units)

Scalability is a key issue for optimization algorithms, especially for solving spatial allocation problems for which the number of decision variables is determined by the size and heterogeneity of the study area, the spatial resolution, and the precise nature of the optimization problem (e.g. the number of optional land use classes). There is empirical evidence that efficacy and efficiency of multi-objective metaheuristics decrease for large numbers of decision variables (Antonio and Coello Coello, 2016; Durillo et al., 2010). Constraint-handling based on repair mechanisms such as GC-GM comes at additional computational cost which might increase excessively with individual size (i.e. in our case with the number of spatial units considered). Our experiment therefore included performance tests with a varying number of spatial units, ranging from 41 to 400. The lowest complexity (n = 41) was represented by clustering neighboring cells of the same land use type (land use map, Fig. 4a) into patches. The number of spatial units was increased to 100 in a cell-level optimization using the same map. We further considered individuals of size 400 by increasing the spatial resolution by factor two in a cell-level optimization. In each case, the initial land use distribution (Figure 4a) was retained.

Number of objectives

The performance of Pareto-based algorithms also generally decreases with increasing number of objectives due to the inefficiency of the Pareto relation in high-dimensional spaces (n > 3, Knowles and Corne, 2007; Li et al., 2015; López Jaimes and Coello Coello, 2015). We tested the applicability of CoMOLA for maximizing two (crop yield and habitat heterogeneity), three (crop yield, habitat heterogeneity, and forest species richness), and four objectives (crop yield, habitat heterogeneity, forest species richness, and water yield).

Strength of constraints

One can assume that the effect of constraint-handling methods on optimization performance depends on the strength of the pre-defined constraints. We therefore tested two different constraint scenarios. The first scenario (CONS_I) took only a set of transition rules into consideration, where (1) existing cropland could be converted either into a different type (intensity) of cropland or into pasture or forest, (2) existing pasture could be converted into forest, but not the other way around, meaning that (3) existing forest had to remain at the same location, while (4) urban areas were not allowed to change at all. The second scenario (CONS_II) used the same transition rules, but also took into account the permissible total area for different land use classes, ranging from 10 to 30 % for forest, from 10 to 20 % for pasture and from 10 to 25 % for cropland 1 and 5, respectively. In addition, we applied CoMOLA without constraints using the original NSGA-II algorithm without any modification. The unconstrained optimization runs served as a reference for the maximum solution space regardless of their realism.

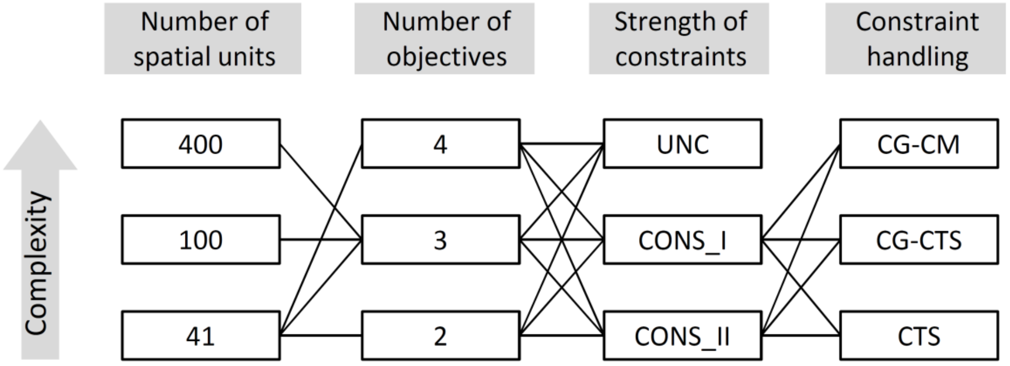

Figure 5: Experimental design including 35 different settings, each setting was repeated in ten independent runs. UNC stands for an optimization problem without constraints, CONS_I includes transition constraints, CONS_II includes both transition and total area constraints. CG-CTS and CTS refer to constraint tournament selection with and without a feasible initial population, respectively, while CG-CM refers to repair mutation.

General settings and performance evaluation

For each optimization run, we used the same population size (300), number of generations (300), crossover rate (0.9), mutation rate (1/individual size), and included boundary solutions (seeds) within the initial population.

In addition to the number of feasible non-dominated solutions, we used the hypervolume metric as proposed by Zitzler and Thiele (1999) to evaluate each of the different solution sets. This widely accepted multi-objective performance metric measures both convergence and diversity on a single scale without requiring the knowledge of the true Pareto front for comparison (Jiang et al., 2014). Hypervolume represents the volume in the objective space that is dominated by the set of feasible solutions given a certain reference point (such as the origin of coordinates). Higher values of the hypervolume hence indicate that the solutions are closer to the true Pareto front and, at the same time, that they are more evenly scattered in the objective space (Jiang et al., 2014). We normalized the fitness values for each objective by dividing by their unconstrained theoretical maximum value (section 3.1), which resulted in values for each objective between 0 and 1. This ensured no objective was overrepresented in the calculation of hypervolumes. Moreover, using the unconstrained theoretical maximum as a fixed normalization factor allowed comparability among different constraint scenarios and constraint-handling methods. The R package ‘mco’ (Mersmann, 2014) was applied, where the origin of coordinates was defined as reference point.

We used wall time (or real time; includes the CPU and the kernel time) as a measure of computation time. Since we are dealing with stochastic algorithms, we have executed ten independent runs for each problem instance which led to a total of 350 optimization runs. Each optimization run was performed on a Linux cluster using 20 cores in parallel. Routines to run the models in parallel are included in the tool.

4. Results & Discussion

4.1 Plausibility and visualization of non-dominated solutions

As expected, the front of non-dominated solutions for the unconstrained optimization problem completely dominated the identified front for a constrained problem (see two-objective optimization example in Fig. 6) – if land use composition and land use transitions are unconstrained the optimizer is able to explore a larger search space. However, not a single solution of the unconstrained optimization run satisfied our hypothetical constraints on land use transition and total area for the individual land use classes, pointing out the (obvious) need for constraint-handling in our example application. Using repair mutation, in contrast, ensured feasibility of any single individual generated during the optimization.

The shape of the fronts implies an almost linear negative relationship (trade-off) between the two objectives, which is plausible since increasing habitat heterogeneity impairs crop production, as defined in our models. The shape of the true Pareto fronts might be more convex due to the impact of soil fertility on crop yield. However, from analyzing the example maps (also shown in Fig. 6) it is fair to assume that the optimization runs identified at least near-optimal solutions. This assumption seems plausible, especially because highest-intensity cropland has been pre-dominantly allocated in the right part of the maps where soil fertility is highest (cf. Fig. 4b), leaving the mosaic of extensive land use classes responsible for highest habitat heterogeneity in the left part of the maps.

Results for the constrained optimization runs considering three and four objectives were plausible in the same sense, see Figures 7 and 8. Forest species richness as a third objective showed a clear trade-off with crop yield but not with habitat heterogeneity. Because of the pre-defined constraints (allowing forest not to exceed a total area of 30 %), there was no single maximum solution for forest species richness but a whole set of solutions, ranging over large parts of the objective space including the maximum of habitat heterogeneity. This is why the marked maximum solutions for forest species richness coincided with the maximum solutions for habitat heterogeneity in Figures 7 and 8.

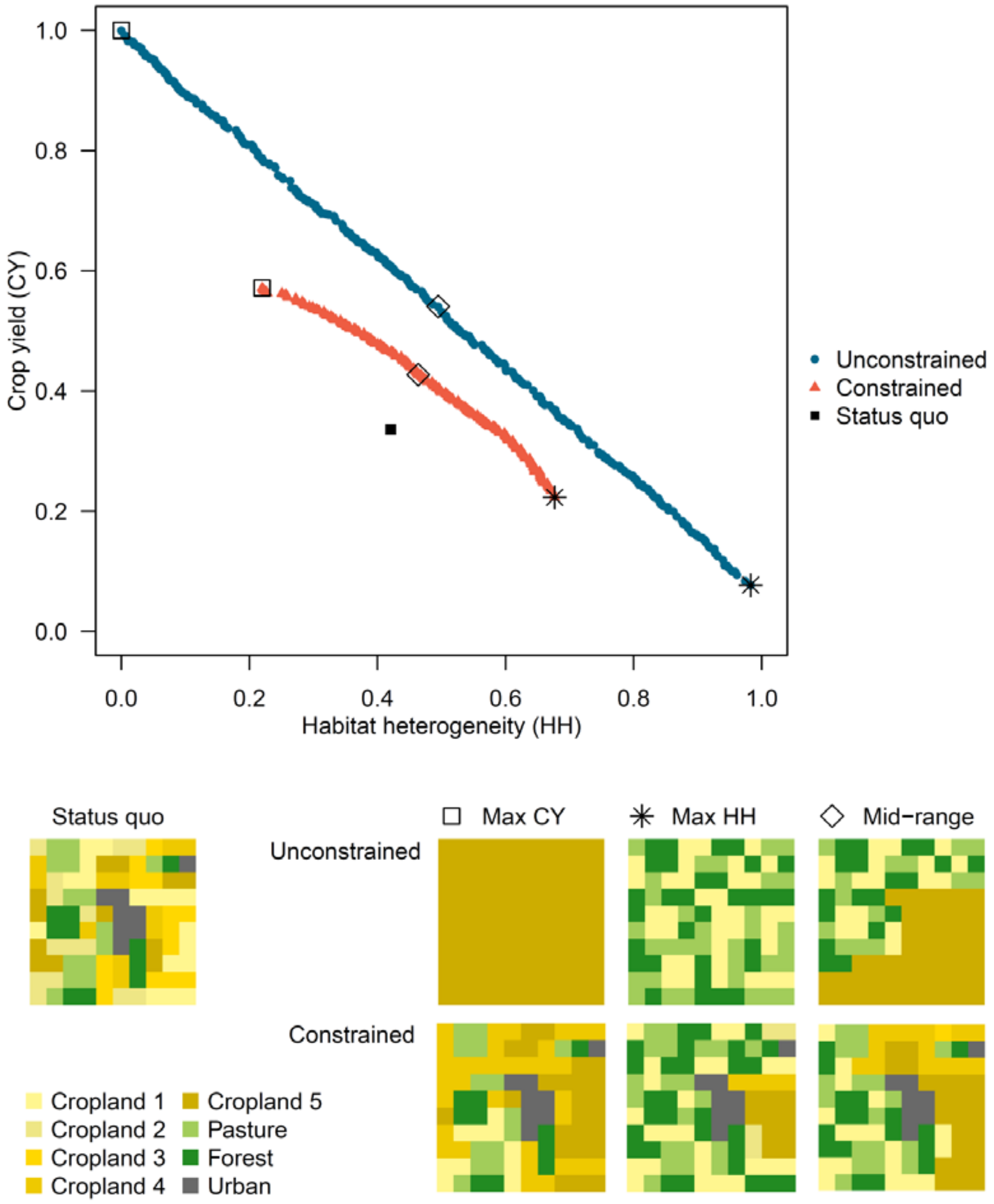

Figure 6: Non-dominated solutions of an unconstrained and a constrained patch-level two-objective optimization run compared to the status quo land use with illustration of respective extreme and mid-range solutions. *Max CY* is the solution with maximal crop yield and *Max HH* the solution with maximal habitat heterogeneity. Values were normalized by unconstrained theoretical maximum values. The constrained optimization was carried out using repair mutation considering both transition and area constraints.

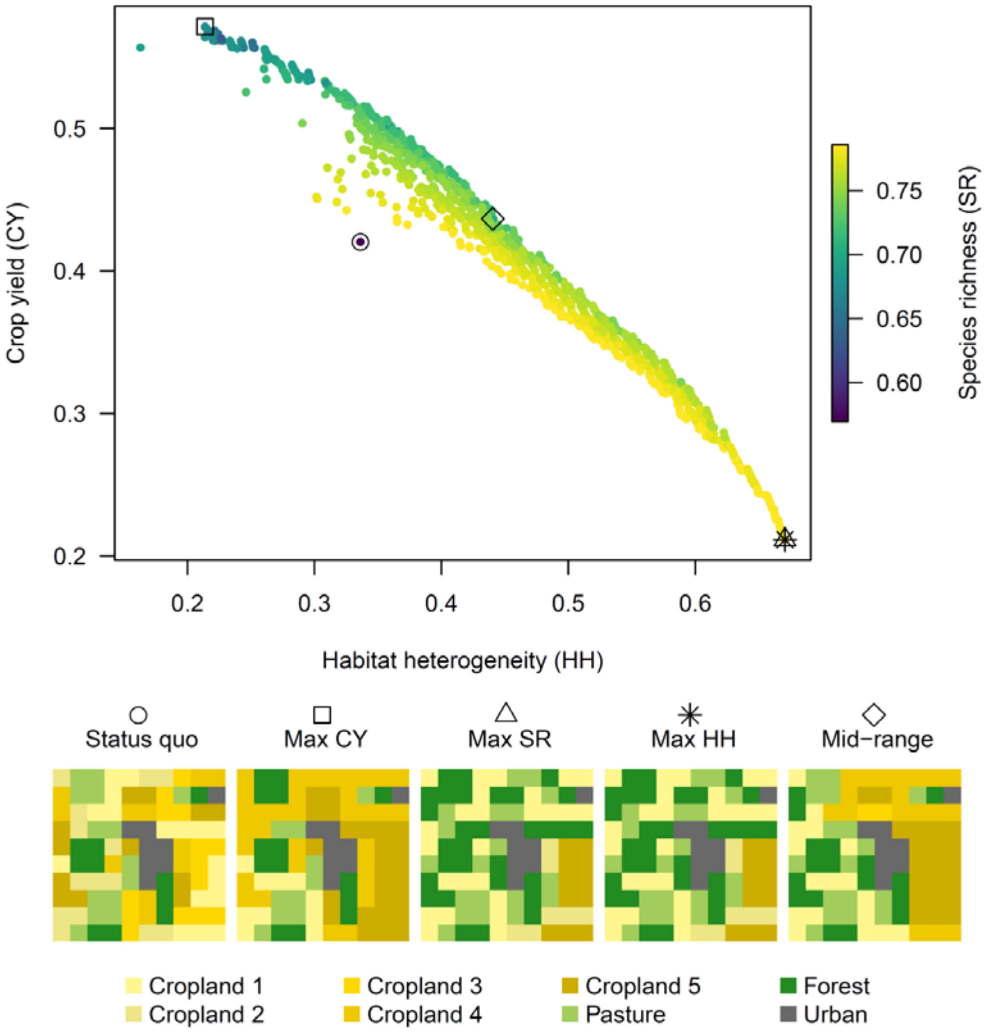

Figure 7: Non-dominated solutions of a constrained patch-level three-objective optimization run using repair mutation considering both transition and area rules compared to the status quo land use with illustration of extreme and mid-range solutions. *Max CY* is the solution with maximal crop yield, *Max SR* the solution with maximal species richness and *Max HH* the solution with maximal habitat heterogeneity. Values were normalized by the theoretical (unconstrained) optimum values for each objective.

Water yield as fourth objective (Fig. 8) favored low-intensity cropland and thus showed a clear trade-off with forest species richness, similar to the crop yield – forest species richness relationship. However, the relationship between water yield and both crop yield and habitat heterogeneity at the Pareto frontier was not that clear. Moderate crop yield and habitat heterogeneity occurred at maximum water yield, caused by the highest share of low-intensity cropland for this solution.

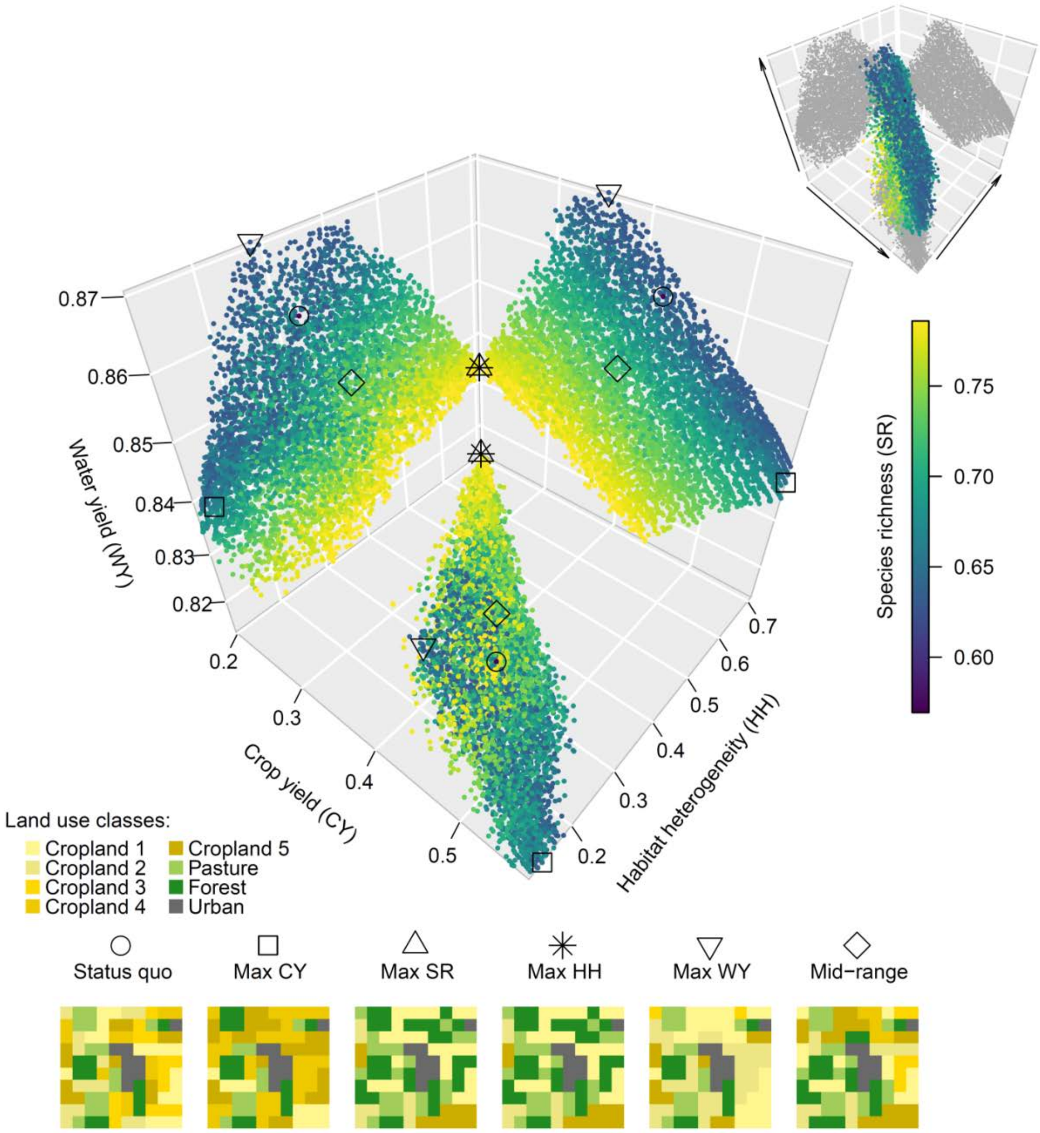

Figure 8: Non-dominated solutions of a constrained patch-level four-objective optimization run using repair mutation considering both transition and area rules compared to the status quo land use with illustration of extreme and mid-range solutions. *Max CY* is the solution with maximal crop yield, *Max SR* the solution with maximal species richness, *Max WY* the solution with maximal water yield and *Max HH* the solution with maximal habitat heterogeneity. Values were normalized by the theoretical (unconstrained) optimum values for each objective. Note that the 3D solution cloud (shown in the top right but left out in the main plot) is projected onto three orthogonal planes (i.e. the solutions are shown in 2D from three different perspectives). Colors indicate the forth objective (species richness). The small figure at the top right shows the plot in real 3D with projections on the three axes shown in grey.

What becomes obvious from Figure 8 is that visualizing a four-dimensional solution space in just one graph is extremely challenging, in particular for a high number of solutions. It is known that the number of non-dominated solutions increases enormously with the number of objectives (Deb, 2001), causing incomparability of solutions and inefficiency of Pareto-based sorting schemes (e.g. Li et al., 2015). This is why NSGA-II is usually not used for more than three or four objectives. The number of solutions could have been reduced by applying an ε-dominance analysis (Laumanns et al., 2002). In such a case, colored 3D scatterplots or even 2D scatterplots with color and point size as third and fourth dimension, respectively, seem to be adequate (cf. Lautenbach et al., 2013). Our suggestion of projecting a colored 3D Pareto cloud onto three orthogonal planes of a cube while leaving out the actual 3D plot (as shown in Fig. 8) might be a possibility to visualize a high number of 4D solutions. However, finding proper techniques for visualizing high-dimensional solutions is a research field on its own (cf. Chiu et al., 2009; He and Yen, 2016).

4.2 Optimization performance

The performance of CoMOLA was determined by (1) the selection of the constraint-handling method, (2) the definition of the optimization problem in terms of constraints, and (3) the size of an individual (i.e. the number of spatial units involved). In general, we found that repair mutation (CG-CM) clearly outperformed both methods based on constraint tournament selection (CG-CTS and CTS). This was indicated by both, a larger number of feasible individuals in the final set of non-dominated solutions and larger hypervolumes dominated by the feasible solutions (cf. Fig. 9 and, additionally, Fig. A.2 in the Supplementary for considering two and four objectives). Our results thus imply that for discrete optimization problems repair mechanisms might be the preferable option to handle hard constraints, as also found by Zydallis and Lamont (2001) for non-spatial problems.

The differences in optimization performance among constraint-handling methods increased with the strength of constraints and the size of an individual. While a considerable amount of feasible solutions was found with all methods for the less restrictive constraint scenario (CONS_I = transition constraints only), the number of feasible solutions was substantially smaller for the more restrictive scenario considering both, transition, and total area constraints (CONS_II). For CONS_II in combination with large individual sizes (100 and 400), not even a single feasible individual was included in the final set of non-dominated solutions when using constraint tournament selection, which is why hypervolumes were assigned a value of zero in this case (Fig. 9). In contrast, our proposed repair algorithm ensured feasibility and uniqueness of all individuals (n = 90.300) generated during an optimization run.

The improved performance of repair mutation, however, came at a cost. As Coello Coello et al. (2007) stated “repairing infeasible solutions in order to make them feasible […] may be computationally expensive”. We found that the runtime for CG-CM increased substantially with the strength of constraints (Fig. 9).

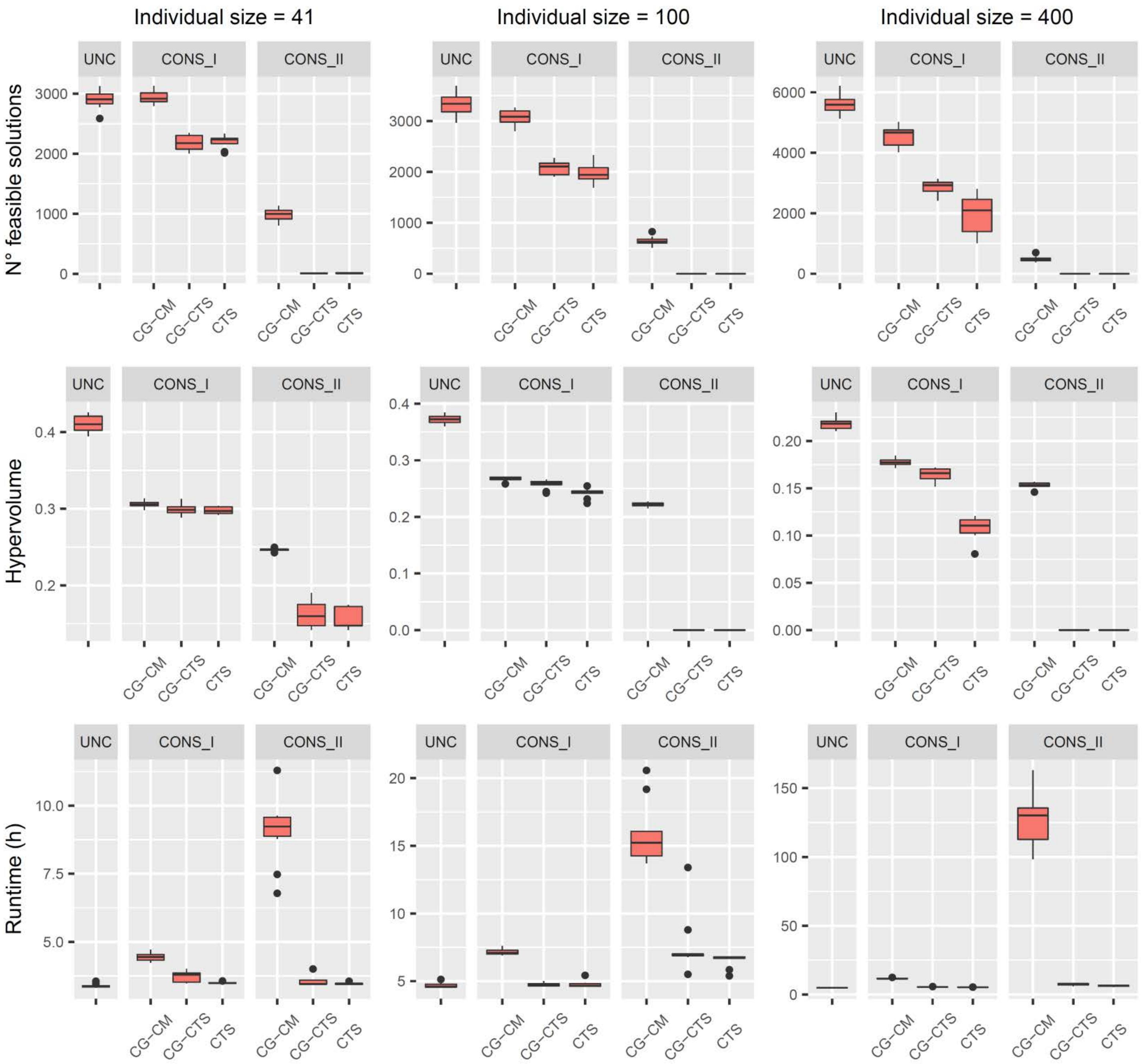

Figure 9: Performance values of three-objective optimization runs for different individual sizes (i.e. number of spatial units considered) shown as boxplots of ten repetitions. UNC stands for an optimization problem without constraints, CONS_I includes transition constraints, CONS_II includes both transition and total area constraints. CG-CTS and CTS refer to constraint tournament selection with and without a feasible initial population, respectively, while CG-CM refers to repair mutation.

The smaller the feasible solution space, the more effort is required to generate adequate individuals. Compared to the other methods that we tested, CG-CM required three times more computation time in the more restrictive constraint scenario with the smallest size of individuals, while the difference was not as pronounced for the less strict constraint scenario. Further, the difference in runtime increased enormously with larger size of individuals, in our case up to factor 20 when handling individuals of size 400 compared to 41. In contrast, runtime with CTS-based constraint-handling increased only marginally (Fig. 9). However, CTS and CG-CTS might be of little use for larger individuals and strong constraints as they may result in only infeasible solutions, as discussed above.

The achieved level of hypervolumes and thus the optimization performance decreased substantially with individual size, c.f. Figure 10. This is also indicated by the different slopes of the hypervolume curves in case of repair mutation. Hypervolume curves converging laterally towards a certain threshold, as it is the case for the smallest individual size, might suggest (but not guarantee) a good approximation of the true Pareto front. Curves progressing with a distinct slope until the end of the optimization process (after 300 generations), as shown for larger individual sizes, clearly indicate further potential of progress.

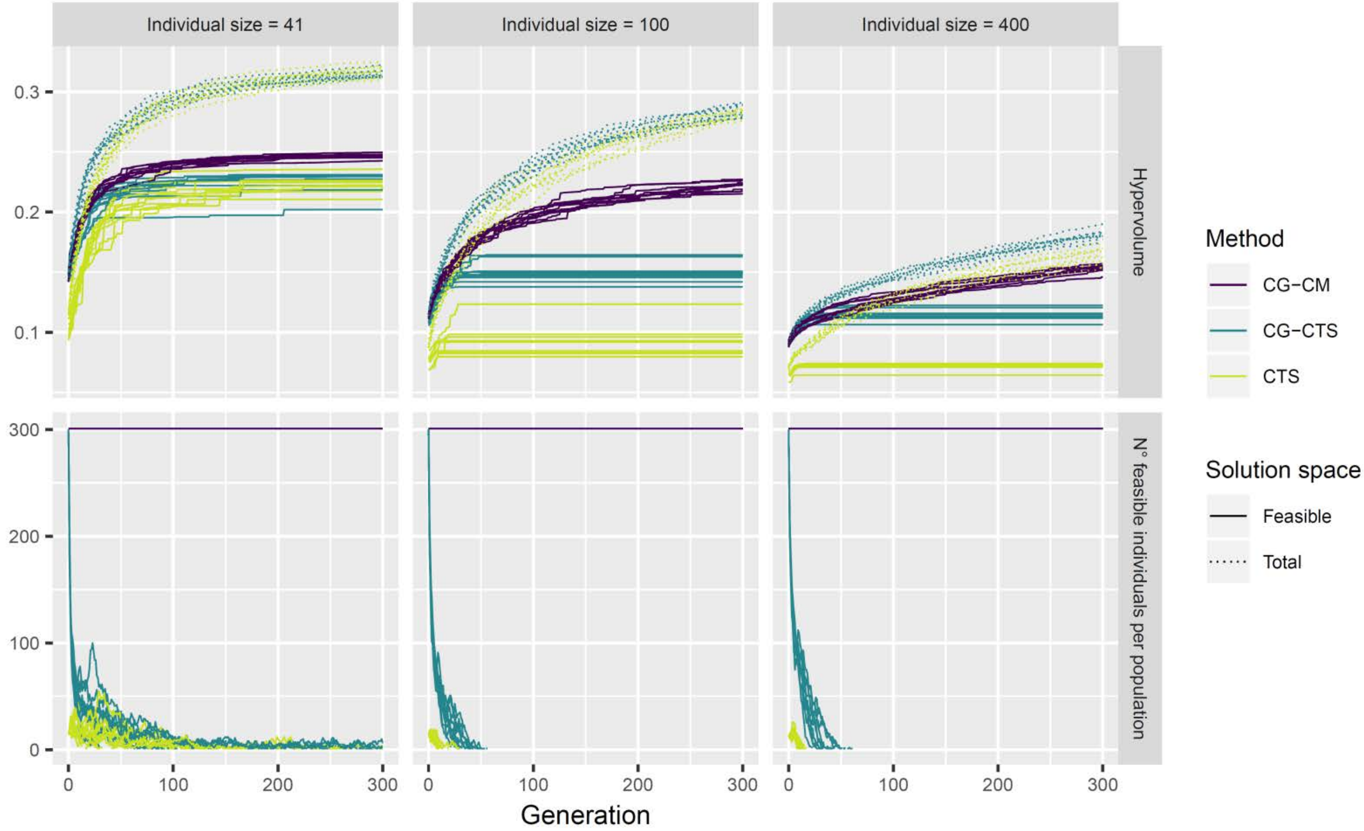

Figure 10: Evolution of the hypervolume (top row) and number of feasible individuals (bottom row) over all generations of three-objective optimization runs solving the more restrictive constraint scenario (CONS_II, including both transition and total area constraints) with different constraint-handling methods for different individual sizes, each in ten repetitions. CG-CTS and CTS refer to constraint tournament selection with and without a feasible initial population, respectively, while CG-CM refers to repair mutation. Examples of respective non-dominated fronts are shown in Fig. A.3 in the Supplementary.

For constrained tournament selection, the hypervolumes of the entire solution space (i.e. all solutions, with no guarantee of feasibility) were considerably larger than the hypervolumes achieved with repair mutation. However, in the CTS-based optimization with larger individual sizes (100 and 400), infeasible individuals completely displaced the feasible ones after only a few generations. This is why here the hypervolume progress of the feasible solution space, as reconstructed a-posteriori from the optimization archive, comes to an abrupt end (see middle and right panel of Fig. 10). Although constrained tournament selection generally favors feasible individuals over infeasible ones, the displacement of feasible individuals might be caused by crossover and mutation, which presumably tended to violate the given constraints. The types of constraints considered here, especially the minimum and maximum allowable total area of a specific land use class, hampered the generation of feasible offspring, even if feasible parents were selected for mating. Offspring individuals in spatial optimization are always a mosaic of subsets of parent maps and even though the parent maps are feasible according to the total area constraints, that does not have to be true for their subsets and even less so for the product of recombination. Constraint-handling methods focusing only on the selection process for mating, such as CTS, may therefore not be adequate for spatial optimization problems that include hard total area constraints.

The decrease in the number of feasible solutions (with increasing number of generations) for CG-CTS and CTS was stronger for optimization problems with larger individuals because the likelihood of a constraint-violating land use change increases with the number of spatial units. The difference in terms of feasible individuals per population for CG-CTS and CTS was marginal between individual size 100 and 400. In both cases, the number of feasible individuals in a population dropped to zero after around 50 and 20 generations for CG-CTS and CTS, respectively (Fig. 10 and, for a closer look on the first 75 generations, Fig. A.4 in the Supplementary). In case of CG-CTS, all 300 individuals of the initial population (generation 0) were forced to feasibility due to the constraint-controlled genome generation (CG) algorithm. In CTS, the optimization started with only 14 feasible individuals representing the boundary seeds which were included in each experiment. The higher number of feasible individuals in the initial population explains why hypervolumes were larger for CG-CTS compared to CTS. The benefit of accounting for feasibility in the initial population was also previously reported by other studies (e.g. Datta et al., 2012; Haubelt et al., 2005). For the smallest individual size, feasible individuals only became nearly extinct for CG-CTS and CTS. From around generation 75, the number of feasible individuals plateaued at a low level, irrespective from their initial number (300 in CG-CTS vs. 14 in CTS). This explains why differences in hypervolume evolution between CG-CM and the constraint tournament selection approaches were less pronounced for the smallest individual size.

The loss of performance with increasing size of individuals was observed for all settings, independent from whether constraints were considered or which constraint-handling method was used (cf. Figs. 9 and 10). Hypervolumes for CG-CM decreased on average by 10% and 38% when increasing the individual size from 41 to 100 and 400, respectively. The average performance loss was similar for the unconstrained optimization runs with 10% and 46%.

Both, increases in runtime and decreases in performance with increasing individual size were caused by the exponential growth of the number of possible combinations (N) that equaled K^J in optimization runs without constraints, where K is the number of possible land use options and J is the number of spatial units (patches or grid cells). This amounted to $N \in \{8^{41},\ 8^{100},\ 8^{400}\}$ possible options for each of the different individual sizes tested, respectively. If constraints are defined, N is smaller, but cannot be predicted because the number of possible land use options varies across the spatial units and can be even variable for a single spatial unit when considering total area constraints.

Despite the performance loss for larger individual sizes, CoMOLA with its implemented repair algorithm was able to detect a multitude of feasible solutions that significantly outperformed the status quo land use simultaneously for all objectives, and such a set of solutions might still have utility in real-world case studies (cf. Fig. A.5 in the Supplementary). This was also shown by Verhagen et al. (2018) who applied CoMOLA for 277 pasture farms and 141 fruit orchards, which equals a total individual size of 418, in the Kromme Rijn area (~219 km²) in the Netherlands to optimally allocate different agri-environment measures under consideration of both land use transition and total area constraints. The authors aimed to simultaneously maximize the yearly fruit yield profit, the potential habitat of the great crested newt (*Triturus cristatus*), and the aesthetic value of the landscape, while minimizing the loss in pasture production. The study also showed the potential of CoMOLA as a model integration and optimization framework to provide insights into the functional trade-offs associated with different management options and their respective potential to increase landscape multifunctionality. By using CoMOLA, Verhagen et al. (2018) further showed how landscape optimization approaches can be integrated into spatial planning and inform policy design and implementation.

5. Conclusions

This paper introduces CoMOLA as a tool for constrained multi-objective optimization of land use allocation and analyzed its functionality and performance using virtual data and simplified models under consideration of different levels of complexity. We found that CoMOLA is able to identify near-optimal solutions for up to four objectives and the smallest individual size tested (n = 41). If solutions close to the true Pareto front are sought, the individual size in an unconstrained optimization should therefore

not exceed a value of 120 (40) for two (eight) possible land use options, as this amounts to a potential decision space of 2^{120}, similar to our smallest individual size experiment. Since constraints reduce the potential decision space, these limits might be higher for constrained optimization runs.
This analysis found that further increasing the individual size can lead to a substantial loss of optimization performance. By using our approach of a constraint-controlled genome generation and repair mutation, CoMOLA was still able to identify multiple solutions that were significantly better than the status quo land use distribution in all objectives simultaneously. Thus, if it is sufficient to improve the status quo of a real-world case study without approaching true Pareto optimality, individuals with a size of several hundred might still be viable for decently sized computational resources. For an application at larger scales, it might become necessary to aggregate spatial units. Apart from that, time consuming simulation models can increase computational time dramatically if they have to be called repeatedly during the optimization. The use of meta-models (also called surrogate model) is therefore encouraged where feasible (Maier et al., 2014).

Repair mutation was found to be advantageous over constraint tournament selection, at least for spatial optimization problems with transition and total area constraints. Repairing infeasible individuals is, however, computationally intensive and can thus require longer runtimes, depending on the strength of the pre-defined constraints. Each of our constraint scenarios included hard constraints, which we wanted to be satisfied by all means. In case a land use optimization problem includes only soft constraints, such as for example total area rules which should be matched only approximately, penalizing constraint violation during tournament selection could be a better option, preferably with a feasible initial population.

CoMOLA is able to handle basic constraints on land use transition and the total area for each land use class. Future developments should focus on the implementation of more sophisticated constraints such as spatial relationships between land use types or compactness of land use patches (e.g. Eikelboom et al., 2015). To increase the efficiency of CoMOLA, it might be promising to implement and test further constraint-handling methods, such as crossover operators explicitly accounting for feasibility as proposed by García et al. (2017) and Karakostas and Economou (2014) or the repair mutation methods suggested by Schwaab et al. (2018b). It might also be worth utilizing more advanced many-objective evolutionary algorithms (Deb and Jain, 2014; Li et al., 2015; Rui et al., 2013) to overcome problems related to inefficient Pareto-based ranking and thus to include more than four objectives in future applications.

CoMOLA provides a generic framework for combining knowledge and models from various disciplines so that the best possible trade-offs between different landscape-related objectives can be explored. Stakeholders can participate in such a framework by defining optimization objectives and constraints as well as by critically evaluating the set of Pareto-optimal solutions to derive recommendations for desired land use and conservation strategies. It therefore has a high relevance and application potential for studies focused on spatial planning, integrated land and water management and any other field of research that aims to optimize spatial patterns for multiple objectives.

Acknowledgements

This research was funded through the 2013/2014 BiodivERsA/FACCE-JPI joint call, with the national funder BMBF - German Federal Ministry of Education and Research (Project "TALE – Towards multifunctional agricultural landscapes in Europe: Assessing and governing synergies between food production, biodiversity, and ecosystem services", grant 01 LC 1404 A). Author contributions: M.S., A.C., R.S. and M.V. designed the research; C.P., M.S., S.L., C.S. and A.K contributed with programming and software documentation, M.S. performed the research. All authors contributed in writing the paper.

References

Allen, R., Pereira, L., Raes, D., Smith, M., 1998. Crop evapotranspiration: guide-lines for computing crop water requirements. FAO Irrigation and DrainagePaper No. 56. FAO, Rome, Italy.

Antonio, M.L., Coello Coello, C.A., 2016. Decomposition-Based Approach for Solving Large Scale Multi-objective Problems. In: Handl, J. et al. (Eds.), Parallel Problem Solving from Nature – PPSN XIV: 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings. Springer International Publishing, Cham, pp. 525-534. DOI:10.1007/978-3-319-45823-6_49

Bennett, D.A., Xiao, N., Armstrong, M.P., 2004. Exploring the Geographic Consequences of Public Policies Using Evolutionary Algorithms. Annals of the Association of American Geographers, 94(4): 827-847. DOI:10.1111/j.1467-8306.2004.00437.x

Boussaïd, I., Lepagnot, J., Siarry, P., 2013. A survey on optimization metaheuristics. Information Sciences, 237: 82-117. DOI:10.1016/j.ins.2013.02.041

Brunner, S.H., Huber, R., Grêt-Regamey, A., 2016. A backcasting approach for matching regional ecosystem services supply and demand. Environmental Modelling & Software, 75: 439-458. DOI:https://doi.org/10.1016/j.envsoft.2015.10.018

Cao, K. et al., 2011. Spatial multi-objective land use optimization: extensions to the non-dominated sorting genetic algorithm-II. International Journal of Geographical Information Science, 25(12): 1949-1969. DOI:10.1080/13658816.2011.570269

Cao, K., Huang, B., Wang, S., Lin, H., 2012. Sustainable land use optimization using Boundary-based Fast Genetic Algorithm. Computers, Environment and Urban Systems, 36(3): 257-269. DOI:http://dx.doi.org/10.1016/j.compenvurbsys.2011.08.001

Chikumbo, O., Goodman, E., Deb, K., 2015. Triple Bottomline Many-Objective-Based Decision Making for a Land Use Management Problem. Journal of Multi-Criteria Decision Analysis, 22(3-4): 133-159. DOI:doi:10.1002/mcda.1536

Chiu, P.-W., Naim, A.M., Lewis, K.E., Bloebaum, C.L., 2009. The hyper-radial visualisation method for multi-attribute decision-making under uncertainty. International Journal of Product Development, 9(1,2,3): 4-31. DOI:https://doi.org/10.1504/IJPD.2009.026172

Coello Coello, C.A., 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering, 191(11–12): 1245-1287. DOI:http://dx.doi.org/10.1016/S0045-7825(01)00323-1

Coello Coello, C.A., Lamont, G.L., van Veldhuizen, D.A., 2007. Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg.

Cord, A.F. et al., 2017. Towards systematic analyses of ecosystem service trade-offs and synergies: Main concepts, methods and the road ahead. Ecosystem Services. DOI:https://doi.org/10.1016/j.ecoser.2017.07.012

Datta, D. et al., 2007. Multi-Objective Evolutionary Algorithm for Land-Use Management Problem. International Journal of Computational Intelligence Research, 3(4): 371-384.

Datta, D., Malczewski, J., Figueira, J.R., 2012. Spatial Aggregation and Compactness of Census Areas with a Multiobjective Genetic Algorithm: A Case Study in Canada. Environment and Planning B: Planning and Design, 39(2): 376-392. DOI:doi:10.1068/b38078

Deb, K., 2000. Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithm. Computer Methods in Applied Mechanics and Engieering 186, 311-338, 186, 311-338 pp. DOI:10.1016/S0045-7825(99)00389-8

Deb, K., 2001. Multi-Objective Optimization Using Evolutionary Algorithms Wiley, Chichester, U.K.

Deb, K., 2014. Multi-objective Optimization. In: Burke, E., Kendall, G. (Eds.), Search Methodologies. Springer, Boston, MA.

Deb, K., Jain, H., 2014. 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. DOI:10.1109/TEVC.2013.2281535

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2): 182-197. DOI:10.1109/4235.996017

Duh, J.-D., Brown, D.G., 2007. Knowledge-informed Pareto simulated annealing for multi-objective spatial allocation. Computers, Environment and Urban Systems, 31(3): 253-281. DOI:http://dx.doi.org/10.1016/j.compenvurbsys.2006.08.002

Durillo, J. et al., 2010. A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems. IEEE Transactions on Evolutionary Computation, 14(4): 618-635. DOI:10.1109/TEVC.2009.2034647

Eikelboom, T., Janssen, R., Stewart, T.J., 2015. A spatial optimization algorithm for geodesign. Landscape and Urban Planning, 144: 10-21. DOI:http://dx.doi.org/10.1016/j.landurbplan.2015.08.011

Fontana, V. et al., 2013. Comparing land-use alternatives: Using the ecosystem services concept to define a multi-criteria decision analysis. Ecological Economics, 93: 128-136. DOI:https://doi.org/10.1016/j.ecolecon.2013.05.007

Friedrich, T., Wagner, M., 2015. Seeding the initial population of multi-objective evolutionary algorithms. Appl. Soft Comput., 33(C): 223-230. DOI:10.1016/j.asoc.2015.04.043

García, G.A., Rosas, E.P., García-Ferrer, A., Barrios, P.M., 2017. Multi-Objective Spatial Optimization: Sustainable Land Use Allocation at Sub-Regional Scale. Sustainability, 9(6): 927.

Garrett, A., 2012. Inspyred: Bio-inspired Algorithms in Python. https://pypi.python.org/pypi/inspyred.

Groot, J.C.J., Oomen, G.J.M., Rossing, W.A.H., 2012. Multi-objective optimization and design of farming systems. Agricultural Systems, 110: 63-77. DOI:http://dx.doi.org/10.1016/j.agsy.2012.03.012

Haubelt, C., Gamenik, J., Teich, J., 2005. Initial population construction for convergence improvement of MOEAs, Proceedings of the Third international conference on Evolutionary Multi-Criterion Optimization. Springer-Verlag, Guanajuato, Mexico, pp. 191-205. DOI:10.1007/978-3-540-31880-4_14

He, Z., Yen, G., 2016. Visualization and Performance Metric in Many-Objective Optimization. IEEE Transactions on Evolutionary Computation, 20(3): 386-402. DOI:10.1109/TEVC.2015.2472283

Holzkämper, A., Seppelt, R., 2007. A generic tool for optimising land-use patterns and landscape structures. Environmental Modelling & Software, 22(12): 1801-1804. DOI:http://dx.doi.org/10.1016/j.envsoft.2007.02.008

Huang, K., Liu, X., Li, X., Liang, J., He, S., 2013. An improved artificial immune system for seeking the Pareto front of land-use allocation problem in large areas. International Journal of Geographical Information Science, 27(5): 922-946. DOI:10.1080/13658816.2012.730147

Ishibuchi, H., Tsukamoto, N., Nojima, Y., 2008. Evolutionary many-objective optimization: A short review, 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 2419-2426. DOI:10.1109/CEC.2008.4631121

Jain, H., Deb, K., 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an

Adaptive Approach. IEEE Transactions on Evolutionary Computation, 18(4): 602-622. DOI:10.1109/TEVC.2013.2281534

Jiang, S., Ong, Y.S., Zhang, J., Feng, L., 2014. Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization. IEEE Transactions on Cybernetics, 44(12): 2391-2404. DOI:10.1109/TCYB.2014.2307319

Kaim, A., Cord, A.F., Volk, M., 2018. A review of multi-criteria optimization techniques for agricultural land use allocation. Environmental Modelling & Software, 105: 79-93. DOI:https://doi.org/10.1016/j.envsoft.2018.03.031

Kanter, D.R. et al., 2016. Evaluating agricultural trade-offs in the age of sustainable development. Agricultural Systems. DOI:http://dx.doi.org/10.1016/j.agsy.2016.09.010

Karakostas, S., Economou, D., 2014. Enhanced multi-objective optimization algorithm for renewable energy sources: optimal spatial development of wind farms. International Journal of Geographical Information Science, 28(1): 83-103. DOI:10.1080/13658816.2013.820829

Knowles, J., Corne, D., 2007. Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (Eds.), Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 757-771. DOI:10.1007/978-3-540-70928-2_57

Laumanns, M., Thiele, L., Deb, K., Zitzler, E., 2002. Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. Evolutionary Computation, 10(3): 263-282. DOI:10.1162/106365602760234108

Lautenbach, S., Volk, M., Strauch, M., Whittaker, G., Seppelt, R., 2013. Optimization-based trade-off analysis of biodiesel crop production for managing an agricultural catchment. Environmental Modelling & Software, 48: 98-112. DOI:10.1016/j.envsoft.2013.06.006

Li, B., Li, J., Tang, K., Yao, X., 2015. Many-Objective Evolutionary Algorithms: A Survey. ACM Comput. Surv., 48(1): 1-35. DOI:10.1145/2792984

Liu, J. et al., 2015. Systems integration for global sustainability. Science, 347(6225). DOI:10.1126/science.1258832

Liu, Y., Peng, J., Jiao, L., Liu, Y., 2016. PSOLA: A Heuristic Land-Use Allocation Model Using Patch-Level Operations and Knowledge-Informed Rules. PLoS ONE, 11(6): e0157728. DOI:10.1371/journal.pone.0157728

López Jaimes, A., Coello Coello, C.A., 2015. Many-Objective Problems: Challenges and Methods. In: Kacprzyk, J., Pedrycz, W. (Eds.), Springer Handbook of Computational Intelligence. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1033-1046. DOI:10.1007/978-3-662-43505-2_51

MacArthur, R.H., Wilson, E.O., 1967. The Theory of Island Biogeography. Princeton University Press, Princeton, N.J.

Maier, H.R. et al., 2014. Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. Environmental Modelling & Software, 62: 271-299. DOI:https://doi.org/10.1016/j.envsoft.2014.09.013

Malczewski, J., Rinner, C., 2015. Heuristic Methods, Multicriteria Decision Analysis in Geographic Information Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 145-189. DOI:10.1007/978-3-540-74757-4_6

Marler, R.T., Arora, J.S., 2010. The weighted sum method for multi-objective optimization: new insights. Structural and Multidisciplinary Optimization, 41(6): 853-862. DOI:10.1007/s00158-009-0460-7

Memmah, M.-M., Lescourret, F., Yao, X., Lavigne, C., 2015. Metaheuristics for agricultural land use optimization. A review. Agronomy for Sustainable Development, 35(3): 975-998. DOI:10.1007/s13593-015-0303-4

Mersmann, O., 2014. Multiple Criteria Optimization Algorithms and Related Functions. R package version 1.0-15.1.

Osyczka, A., Krenich, S., 2000. A new constraint tournament selection method for multicriteria optimization using genetic algorithm, Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), pp. 501-508 vol.1. DOI:10.1109/CEC.2000.870338

Polasky, S. et al., 2008. Where to put things? Spatial land management to sustain biodiversity and economic returns. Biological Conservation, 141(6): 1505-1524. DOI:10.1016/j.biocon.2008.03.022

R Core Team, 2016. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Roberts, S.A., Hall, G.B., Calamai, P.H., 2011. Evolutionary Multi-objective Optimization for landscape system design. Journal of Geographical Systems, 13(3): 299-326. DOI:10.1007/s10109-010-0136-2

Rui, W., Purshouse, R.C., Fleming, P.J., 2013. Preference-Inspired Coevolutionary Algorithms for Many-Objective Optimization. Evolutionary Computation, IEEE Transactions on, 17(4): 474-494. DOI:10.1109/TEVC.2012.2204264

Schwaab, J. et al., 2018a. Using multi-objective optimization to secure fertile soils across municipalities. Applied Geography, 97: 75-84. DOI:https://doi.org/10.1016/j.apgeog.2018.06.001

Schwaab, J. et al., 2017. Reducing the loss of agricultural productivity due to compact urban development in municipalities of Switzerland. Computers, Environment and Urban Systems, 65: 162-177. DOI:https://doi.org/10.1016/j.compenvurbsys.2017.06.005

Schwaab, J. et al., 2018b. Improving the performance of genetic algorithms for land-use allocation problems. International Journal of Geographical Information Science, 32(5): 907-930. DOI:10.1080/13658816.2017.1419249

Seppelt, R. et al., 2016. Harmonizing Biodiversity Conservation and Productivity in the Context of Increasing Demands on Landscapes. BioScience, 66(10): 890-896. DOI:10.1093/biosci/biw004

Seppelt, R., Lautenbach, S., Volk, M., 2013. Identifying trade-offs between ecosystem services, land use, and biodiversity: a plea for combining scenario analysis and optimization on different spatial scales. Current Opinion in Environmental Sustainability, 5(5): 458-463. DOI:http://dx.doi.org/10.1016/j.cosust.2013.05.002

Shaygan, M., Alimohammadi, A., Mansourian, A., Govara, Z.S., Kalami, S.M., 2014. Spatial Multi-Objective Optimization Approach for Land Use Allocation Using NSGA-II. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7(3): 906-916. DOI:10.1109/JSTARS.2013.2280697

Stein, A., Gerstner, K., Kreft, H., 2014. Environmental heterogeneity as a universal driver of species richness across taxa, biomes and spatial scales. Ecology Letters, 17(7): 866-880. DOI:doi:10.1111/ele.12277

Stewart, T.J., Janssen, R., van Herwijnen, M., 2004. A genetic algorithm approach to multiobjective land use planning. Computers & Operations Research, 31(14): 2293-2313. DOI:http://dx.doi.org/10.1016/S0305-0548(03)00188-6

Verhagen, W., van der Zanden, E.H., Strauch, M., van Teeffelen, A.J.A., Verburg, P.H., 2018. Optimizing the allocation of agri-environment measures to navigate the trade-offs between ecosystem services, biodiversity and agricultural production. Environmental Science & Policy, 84: 186-196. DOI:https://doi.org/10.1016/j.envsci.2018.03.013

Verstegen, J.A. et al., 2017. How a Pareto frontier complements scenario projections in land use change impact assessment. Environmental Modelling & Software, 97(Supplement C): 287-302. DOI:https://doi.org/10.1016/j.envsoft.2017.08.006

Yoon, Y., Kim, Y.-H., 2014. A Mathematical Design of Genetic Operators on GL(n)(Z(2)). Mathematical Problems in Engineering, 2014: 8. DOI:10.1155/2014/540936

Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation, 3(4): 257-271. DOI:10.1109/4235.797969

Zydallis, J.B., Lamont, G.B., 2001. Solving of discrete multiobjective problems using an evolutionary algorithm with a repair mechanism, Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems. MWSCAS 2001 (Cat. No.01CH37257), pp. 470-473 vol.1. DOI:10.1109/MWSCAS.2001.986213