Available online at www.sciencedirect.com
ScienceDirect
Environmental Modelling & Software 22 (2007) 1801–1804
Environmental Modelling & Software
www.elsevier.com/locate/envsoft

Short communication

A generic tool for optimising land-use patterns and landscape structures

Annelie Holzkämper*, Ralf Seppelt

Helmholtz Centre for Environmental Research – UFZ, Department of Computational Landscape Ecology (CLE), Permoserstrasse 15, D-04318 Leipzig, Germany

Received 31 August 2006; received in revised form 19 February 2007; accepted 28 February 2007
Available online 10 April 2007

Abstract

We present a flexible and easy to use genetic algorithm-based library for optimising the spatial configurations of land-use. LUPOlib, the Land-Use Pattern Optimisation-library, can be applied to a variety of spatial planning problems to derive target-driven scenarios that identify trade-offs between conflicting objectives and solve optimum allocation problems (e.g. allocation of reserve sites or management actions). A major novelty is that spatial changes are performed according to a patch topology that allows to simultaneously integrate changes of different landscape elements (e.g. in agricultural fields and linear changes along corridors). The objective function evaluation is based on a grid representation of the landscape where neighbourhood dependencies like lateral flows or the landscape pattern can explicitly be considered. A parameter file allows the user to control the optimisation, the modelled land-use changes, objective weightings and constraints as well as input data. Only the case study-specific objective function needs to be specified in the source code. LUPOlib has been applied so far in two case studies to find optimum trade-offs between habitat requirements of three different bird species and to analyse cost-effectiveness of management actions for species conservation. The results suggest that LUPOlib can be a useful tool to support management decisions. It could be used as an extension to a GIS and for spatially explicit decision support tools.

Keywords: Genetic algorithm; Land-use pattern optimisation; Conservation management; Landscape metrics

Software availability

Name of Software: LUPOlib 1.0
Programming language: C/C++
Packages used: GAlib 2.4.6 (http://lancet.mit.edu/ga/)
Developers: Annelie Holzkämper
Contact address: Helmholtz Centre for Environmental Research – UFZ, Department of Computational Landscape Ecology, Permoserstrasse 15, D-04318 Leipzig, Germany. E-mail: annelie.holzkaemper@ufz.de
Availability: http://www.ufz.de/index.php?en=4302 (free to use under the GPL, http://www.gnu.org/licenses/gpl.txt)

1. Introduction

In the last few decades the potentials of spatial optimisation in the field of conservation management were discovered. Various software packages have been developed that apply spatial optimisation for finding optimum allocations for reserve sites (e.g. C-Plan by Pressey, Ferrier and Watts, Finkel, 1998; SITES/SPOT/MARXAN, Ball and Possingham, 2000). In this paper we present LUPOlib – Land-Use Pattern Optimisation library, a generic library for grid-based optimisation of spatial landscape configurations with respect to a user-defined optimisation goal. LUPOlib can be applied to a variety of spatial planning problems (e.g. finding trade-off between ecological and economic objectives, optimum allocation of management actions, reserve sites or roads). Like most programs dealing with complex spatial optimisation problems LUPOlib utilises a meta-heuristic search algorithm – a genetic algorithm by

* Corresponding author. Tel.: +49 341 235 3945; fax: +49 341 235 3939.
E-mail address: annelie.holzkaemper@ufz.de (A. Holzkämper).

1364-8152/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.envsoft.2007.02.008

Wall (1996). This optimisation algorithm approaches a global optimum solution in an iterative directed search (Goldberg, 1989). LUPOlib utilises a steady-state genetic algorithm with one-point crossover and flip-mutation as genetic operators. One innovative feature is that changes are performed based on a user-defined patch topology that allows an integration of two different types of land-use changes (e.g. areal and linear). Land use in designated patches (of areas or lines) is changeable, whereas the remaining landscape persists. Within the optimisation the patch topology is represented by two one-dimensional integer arrays (one for areal patches and one for line patches). However, for evaluating the goal function the landscape representation is transformed into a two-dimensional grid. This allows to explicitly consider neighbourhood-dependencies (e.g. for evaluating habitat suitability, landscape metrics or lateral flows of water and nutrients). The objective function needs to be specified by the user to solve a designated spatial optimisation problem. Up to now, the library was applied in two case studies to find optimum land-use patterns with respect to different bird species in parallel and to analyse cost-effectiveness of management actions for species conservation.

2. Description of LUPOlib

LUPOlib is a C/C++ library for the spatial optimisation of spatial land-use patterns. It integrates the C++ library for genetic algorithms GALib (Wall, 1996) and provides an interface between the genetic algorithm and the subject of the optimisation, the spatially explicit landscape. LUPOlib includes functions for reading and writing raster maps, the definition of the landscape representation, algorithms for deriving a user-defined patch topology, for calculating landscape metrics, query functions and a function for applying the optimisation. It operates based on a regular grid M, $M = \{(x,y)|x_{\min} < x < x_{\max}, y_{\min} < y < y_{\max}; x_{\min}, x, x_{\max}, y_{\min}, y, y_{\max} \in \mathrm{IN}\}$

Each grid cell may have different attributes z that either depend on its location (x,y), i.e. $z = f(x,y)$ or on the characteristics of neighbouring cells $N_{x',y'}$, i.e. $z = f(N_{x',y'})$, $N_{x',y'} \subset M$. For example, z_L denotes the land use at location (x,y) in the raster land-use map U. To decrease the computational effort and consider that land-use changes are made in certain decision units (e.g. agricultural fields), LUPOlib transforms the grid into a patch topology. Land-use patches that are potentially subject to change are defined as area units; line units are defined as potential locations for linear land-use changes (e.g. planting of hedges). Linear changes might be relevant to support certain landscape functions (e.g. habitat suitability, erosion and biocontrol). Grid cells of area and line units are changed *en bloc*, as they are assumed to be managed as entire units. Area and line units are identified by unique ID's (Fig. 1). LUPOlib allows deriving ID-maps from U according to user-specified parameters.

The general optimisation task of LUPOlib is to maximise an objective value J by finding an optimum land-use pattern for the units that are modifiable. Thus, U should be identified such that $J(U) \rightarrow \max$. Interactions between neighbouring cells can be considered making J dependent on z that itself dependents on $N_{x',y'}$. The objective function can consist of several objectives that may be weighted to integrate objectives with different units and to specify the relevance of each of the objectives (e.g. habitat suitabilities for different species and costs for changes). One or more constraints can be set (e.g. a budget that limits admissible changes). The objective function allows the incorporation of functions provided by LUPOlib, user-defined functions and even system calls for external programs can be integrated to allow a coupling to existing models.

3. Application of LUPOlib

3.1. Input data

A land-use map U is needed as input map in Arc ASCII raster-format. ID-maps that identify decision units for the optimisation can either be derived from U by LUPOlib or read in as pre-processed ASCII raster-files (e.g. to exclude protected areas from changes). Likewise, further input maps that are needed to evaluate the objective function can be read in.

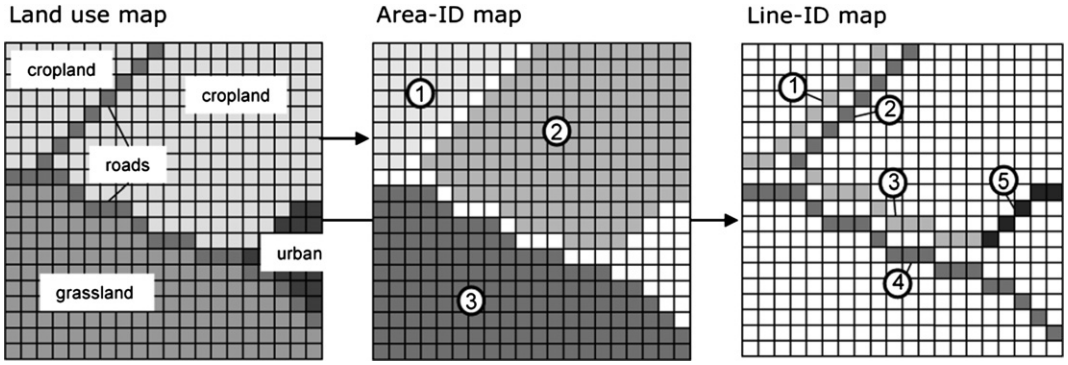

Fig. 1. LUPOlib patch topology: area units and line units are identified in an area- and line-ID map, respectively. Patches of crop- and grassland are assigned to area units; urban areas and roads are excluded from land-use changes. Edge cells of crop- and grassland are assigned to five different line units according to the bordering land-use types.

3.2. Parameter file

A parameter file allows the user to control the genetic algorithm, the possibilities and constraints for land-use change and the derivation of area and line units. Furthermore, input maps are identified and weightings and constraints for the goal function can be defined (Table 1).

3.3. Definition of the objective function

The definition of the objective function sets the optimisation goal. It is problem-specific and thus to be coded by the user. After modification of the objective function the program has to be recompiled. The pseudo-code below shows an example of an objective function to maximise timber harvest profit, while minimising habitat fragmentation given a maximum harvest area defined in a constraint. A weighting is incorporated to allow for an integration of the two objectives (`obj1, obj2`). It applies a user-defined function (`getHarvestProfit()`) and two functions provided by LUPOlib (`getArea(), getEdgeDensity()`).

Table 1
Example parameter file (*see Wall (1996) for documentation of GA parameters)

Parameters	Values	Description
pAU	0.5	Probability for random initialisation of area units
pLU	0.1	Probability for random initialisation of line units
popSize	20	Population size*
pCross	0.6	Crossover probability*
pMut	0.01	Mutation probability*
pRepl	0.25	Probability of replacement*
nGen	100	Maximum number of generations*
pConv	0.99	Convergence criterion*
nConv	100	Convergence parameter*
alleleset_area	4 9 10;	Changeable land-use types
alleleset_line	7;	Linear changes can be made to types
area_change_except	4 10;	No changes are performed to these types
line_change_except	8 9;	No linear changes on these types
line_edge_except	8 9;	No linear changes bordering to these types
AREA_IDMAP	create	Create area-ID map
areaID_cat	4 10;	Assign ID's to patches of these land-use types
neighbourhood	4	Consider 4 nearest neighbours
min_area	1600	Exclude area units <1600 m^2
LINE_IDMAP	line_Idmap	Filename of preprocessed line-ID map
lineID_cat	;	Derive line-ID's for edge-cells of these categories
lineID_except	;	No line-ID's for cells bordering to these categories
min_length	0	Exclude line units < 0 m
weightings	0.5;	Weightings for objective function evaluation
constraints	800;	Constraints for objective function evaluation
landusemap	landuse	Filename of input land-use map
inputmaps	forest_age	Filename of input map for objective evaluation

```
float
objective(landusemap, inputmap,
weighting, constraint){
 float J = 0.000; //objective function value
 float obj1 = 0.000;
 float obj2 = 0.000;
 float con = 0.000;
 //harvested area in landusemap:
 con = getArea(landusemap, harvest);
 if(con <= constraint){
   //harvest profit in landusemap calculated
   //based on input map 'forest_age' (see
   Table 1):
   obj1 = getHarvestProfit(landusemap,
   inputmap);
   //edge density of habitat in landusemap:
   obj2 = getEdgeDensity(landusemap,
   habitat);
   //maximise harvest profit, while
   minimising habitat fragmentation:
   J = weighting*obj1 - obj2;
 }
 return(J);
}
```

A variety of other functions are provided in LUPOlib for calculating different landscape metrics for the whole landscape subset and within a certain radius around a grid cell (e.g. edge density, patch cohesion and largest patch index). All landscape metrics are calculated according to McGarigal et al. (2002). They can be useful, when the optimisation task is to minimise fragmentation or to maximise habitat suitability for species that respond to landscape structure on a territorial scale (Holzkämper et al., 2006).

3.4. Output data

The optimisation algorithm terminates either if the specified convergence criterion or the maximum number of iterations is reached. As model output LUPOlib stores the optimised land-use map in Arc ASCII raster format and a text file with the highest objective values in all processed iterations. ID-maps derived by LUPOlib are also stored with output data.

4. Case studies

LUPOlib has been applied in two different studies so far (Holzkämper et al., 2006; Holzkämper and Seppelt, 2007). The first one dealt with optimising land-use patterns with respect to habitat suitability for three bird species with contrasting habitat requirements. It was investigated where habitat requirements oppose, where they coincide and which

management actions improve habitat suitability for all three species. In a second study, LUPOlib was applied with an ecological-economic goal function to analyse cost-effectiveness of conservation management actions for different bird species on a regional scale by deriving general patterns of Pareto-optimality. These examples demonstrate that LUPOlib can be used to find trade-offs between conflicting management objectives and cost-effective opportunities for ecological improvement.

5. Conclusions and recommendations

Applying a patch topology within the optimisation produced reasonable land-use patterns, as the units of change correspond to the land-use parcels on which decisions are made in reality. LUPOlib can be a useful tool to support conservation management decisions. The library is very flexible and could easily be applied for analysing a variety of other management actions (e.g. concerning land-use intensity or cultivation methods) on different spatial scales with different spatially referenced goal functions (e.g. biodiversity, erosion and leaching of nutrients). Thus, integrated models for decision support, like the ones of Berlekamp et al. (2007) and Dragan et al. (2003), that usually apply scenario analysis could be extended by LUPOlib to create target-driven scenarios. Furthermore, the library might also be incorporated in a GIS such as GRASS. The possibility to integrate areal and linear changes is a major advantage as it allows for optimising the pattern of different landscape elements simultaneously. The program reads and writes maps in Arc ASCII raster format, which is supported by many GIS (e.g. ArcGIS/ArcView and GRASS). It is interoperable, running under Windows and Linux and can easily be used in batch mode.

Acknowledgements

The software presented in this paper uses the GAlib genetic algorithm package by Wall (1996) as well as a routine for patch derivation from Michael Müller, UFZ Leipzig-Halle.

References

Ball, I.R., Possingham, H.P., 2000. MARXAN (V1.8.2): Marine Reserve Design Using Spatially Explicit Annealing, A Manual.

Berlekamp, J., Lautenbach, S., Graf, N., Reimer, S., Matthies, M., 2007. Integration of MONERIS and GREAT-ER in the decision support system for the German Elbe river basin. Environmental Modelling & Software 22 (2), 239–247.

Dragan, M., Feoli, E., Fernetti, M., Zerihun, W., 2003. Application of a spatial decision support system (SDSS) to reduce soil erosion in northern Ethiopia. Environmental Modelling & Software 18 (10), 861–868.

Finkel, E., 1998. Software helps Australia manage forest debate. Science 281, 1789–1791.

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

Holzkämper, A., Lausch, A., Seppelt, R., 2006. Optimizing landscape configuration to enhance habitat suitability for species with contrasting habitat requirements. Ecological Modelling 198, 277–292.

Holzkämper, A., Seppelt, R., 2007. Evaluating cost-effectiveness of conservation management actions in an agricultural landscape on a regional scale. Biological Conservation 136, 117–127.

McGarigal, K., Cushman, S.A., Neel, M.C., Ene, E., 2002. FRAGSTATS: Spatial Pattern Analysis Program for Categorical Maps. University of Massachusetts, Amherst. Available from: <www.umass.edu/landeco/research/fragstats/fragstats.html>

Wall, M., 1996. GALib: A C++ Library of Genetic Algorithm Components. Version 2.4. Available from: <http://lancet.mit.edu/ga/>