A parallel algorithm based on simulated annealing for land use zoning plans

M. Suárez[1], I. Santé[1], F. F. Rivera[2], R. Crecente[1], M. Boullón[1], J. Porta[3], J Parapar[3], and R. Doallo[3]
[1]Land Laboratory, University of Santiago de Compostela, Lugo, Spain
[2]Dept. of Electronics and Computing, University of Santiago de Compostela, Santiago de Compostela, Spain
[3]Computer Architecture Group, University of A Coruña, A Coruña, Spain

Abstract— *There is an increasing demand for tools which support the land use planning process and one of the most complex tasks of this process is the design of a land use zoning map. With this aim an algorithm based on simulated annealing has been designed to optimize the delimitation of land use categories according to suitability and compactness criteria. The high number of plots involved in a land use plan leads to high computational costs. Two parallel versions were implemented. The first one improve the final solution using different parameters in parallel. The second one gets advantage of the spatial parallelism. Results on a real case of study show that the solutions provided by our algorithms are similar to the solution provided by experts, but much faster and with less effort. The parallel versions of the code present good results in terms of the quality of the solution and speed-up.*

Keywords: Land use planning, parallel simulated annealing, GIS

1. Introduction

The development of a land use plan is long and laborious, requiring a great effort on the part of public administrations and technical teams to achieve a good solution. As a result, there is an increasing demand for tools which support the planning process and one of the most complex tasks of this process is the design of a land use zoning map. The design of a land use zoning map can be formulated in terms of an optimization problem in which each plot is allocated to the best category according to certain criteria and constraints. These criteria always include the land suitability for the land uses of a land category (e.g., [1], [2]) and some authors also consider spatial criteria, especially the compactness of the regions allocated to one single category (e.g., [3], [4], [5]). Due to the fact that the number of plots involved in a municipal land use plan is usually large, the search of the optimal solution using algorithms such as integer programming is unfeasible. It is, therefore, necessary to turn to heuristic algorithms capable of achieving near-best solutions in a reasonable time [6] [7] [8]. In particular, good results have been obtained using the simulated annealing technique (e.g., [9], [10], [11], [4], [12], [13]). Most of these algorithms operate on a regular raster grid. Land use zoning based on a regular grid is found to be unrealistic as it may lead to a single-land use plot allocated to several categories or to a group of very different plots allocated to a single category. In addition to this, the planning laws in the study area require land use zoning based on cadastral plots.

The large number of plots involved in a municipal land use plan leads to high computational costs in order to run a number of iterations enough for exploring the complete search space. The use of different parallelization strategies has been considered in order to reduce the execution time and to improve the results of the algorithm. Many proposals for parallelization can be found in the literature [14], [15].

This study proposes a parallel algorithm based on simulated annealing for land use zoning that uses an irregular spatial structure based on a cadastral parcel map. This algorithm was applied to land use zoning in the municipality of Guitiriz, located in Galicia (N.W. of Spain) as a case of study. The paper is structured as follows: Section 2 defines the characteristics of the optimization problem. Section 3 describes the pre-processing stage. Section 4 describes the design of the simulated annealing algorithm. Section 5 is devoted to introduce the implementation of the parallel versions of the algorithm. In section 6 experimental results are discussed. And finally, some conclusions and ideas for future work are given in Section 7.

2. Problem statement

Land use planning laws define a set of land use categories and the restrictions enforced to each category. For some categories, their spatial allocation is completely and uniquely determined by legal restrictions. We will refer to this group of categories as fixed categories. In the case of Galicia, the fixed categories include the water, coast, infrastructure and heritage protection land. The non-fixed categories correspond to the agricultural, forestry, natural space and urban land.

Consequently, two stages can be distinguished in the design of a land use zoning map: the application of law restrictions for the delimitation of fixed categories and the decision making by planners for the allocation of non-fixed categories. For the first stage a pre-processing module has been developed in which the fixed categories are allocated applying the planning laws by means of geometric operations (buffers, intersections, differences...).

In the second stage planners must delimitate the non-fixed categories using their expert knowledge. An heuristic algorithm based on simulated annealing has been designed to facilitate this task. At this point it is important to note that laws and experts advise that the process of spatial allocation should take into account the current boundaries of the existing plots in the municipality, i. e., a plot should not be divided in several parts with different categories. Therefore the problem is to distribute N plots among C different non-fixed categories addressing two objectives, based on experts' criteria: maximization of the overall suitability of the plots to the categories allocated to them and maximization of the compactness (and hence minimization of the fragmentation) of the resultant land use patches. Land use patches are defined as the polygons resulting from the union of plots assigned to the same category. This optimization is subject to the constraints that the total area allocated to each non-fixed category cannot exceed certain minimum and maximum values set by the planner.

The relative importance of both suitability and compactness criteria varies depending on the target land category. For example, the compactness is basic for the forestry land category, whereas in the natural space land the importance of the compactness is low. For this reason the planner must be able to assign different weights to each criterion in each category.

3. Pre-processing stage

The problem requires three types of data which are read in the pre-processing stage: characteristics of each plot, parameters for the allocation of each category and geometric elements to define fixed categories. The characteristics of each plot include its geometry, initial category and a suitability score for each non-fixed category. In addition, the parameters for the allocation of each category include the maximum and minimum area, and the weights for the suitability and compactness criteria for each category.

The elements that define the fixed categories correspond to layers of geometric elements like rivers, roads, archaeological sites. They can delimitate the fixed categories in different ways. The first one is to allocate directly these elements to a specific fixed category. For example, the archaeological sites are included directly in the heritage protection land. This procedure also allows to allocate a category to areas that, because of their singularity, must be delimited by and expert or a specific algorithm. The other issue is the delimitation of a buffer over the geometric elements at a certain distance established by the law. An example is the protection area for roads. The result of these procedures is a map of plots in which the fixed categories are delimited, so the plots allocated to these categories are not considered in the simulated annealing algorithm. The pre-processing stage allows that these calculations, most of them involving computationally expensive geometric operations (e.g. intersections), are run just once. These operations have been implemented using the JTS Topology Suite [16] library for spatial analysis operations and the SEXTANTE framework [17].

Besides, in the pre-processing stage the conditioning of the algorithm input data is carried out. The calculation of the compactness based on land use patches requires the knowledge of the adjacent plots to each one of them, called neighbours and the length of the border that they share. With the aim of speeding up this calculation, in the pre-processing stage a spatial indexing of the plot map is used, so the index query provides the list of plots candidates to be neighbours, thereby reducing the number of processed plots from several tens of thousands to a few tens. Choosing the right data structure to store the neighbours and the length of its borderline is an important issue since this information is accessed often by the algorithm, so it is important to minimize the time to access it. As the number of neighbours of each plot can be different, two unidimensional arrays are used to store the list of neighbours: an array of neighbours and an index array. The *i-th* entry of the index array stores the position in which the first neighbour of the *i-th* plot is stored in the array of neighbours, where $i = 1 \dots N$. The neighbours of each plot are stored consecutively in the array of neighbours. Figure 1 shows an example of these two arrays. In this example neighbours of plot P2 are P1, P3, P6 and P7, and they are stored from position 4 of the array of neighbours. Note that 4 is the value of the second entry in the index array. This structure presents low latency in its access.

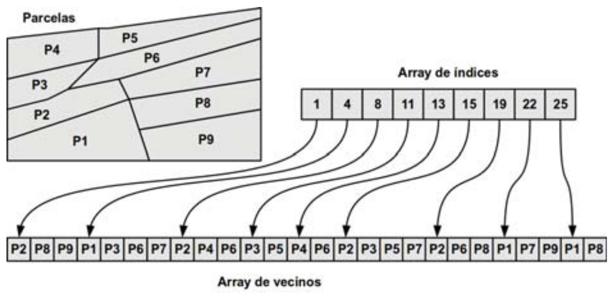

Fig. 1: Arrays used to store the information about the neighbourhood.

4. Simulated annealing algorithm

The simulated annealing algorithm [18] is a heuristic to intensively optimize an objective function ruled by a parameter called temperature T that is used to control the thoroughness of the search for the optimum. The basic procedure is as follows: (1) given the current configuration of the system, a trial configuration is generated by a method that includes some element of chance. (2) The value of the objective function for the trial configuration, E_t, is compared with the value of the objective function for the current configuration,

E_c. If E_t is better than E_c, the trial configuration becomes the current configuration, otherwise the trial configuration is adopted as the next current configuration according to the Boltzmann probability distribution: $e^{(E_c - E_t)/T}$. (3) For each value of temperature, the system is allowed to explore the configuration space for a number of iterations. The value of T is then reduced, so that better E values are favoured and the loop starts from step 1. (4) The algorithm terminates upon satisfaction of some appropriate stop condition.

In our case, at the beginning of the process an initial random solution is generated that satisfies the constraints of maximum and minimum area for each category.

4.1 The objective function

The objective function E combines two subobjectives: maximization of land suitability and maximization of compactness. These subobjectives are combined linearly:

$$E = W_c \times compactness + W_s \times suitability \qquad (1)$$

where W_s and W_c are defined by the planner and are normalized so that the summation of both weights must be 1. The subobjective functions are normalized to the range [0, 1].

Suitability is calculated as the weighted average of the suitability for each category. Suitability for a category is obtained from the average of the suitability of the plots allocated to that category, weighted by the area of each plot and normalized by the total area assigned to the category:

$$Suitability = \sum_{i=1}^{C} w_i \left(\frac{\sum_{j=1}^{N} S_{ij} \times a_{ij}}{\sum_{j=1}^{N} a_{ij}} \right) \qquad (2)$$

where w_i is the weight of the *i-th* category, N_i is the number of plots allocated to the *i-th* category, s_{ij} is the suitability of the *j-th* plot allocated to the *i-th* category, and a_{ij} is the area of the *j-th* plot allocated to the *i-th* category.

Compactness can be defined in different ways. In our proposal two different functions are considered: one based on patches, which are groups of adjacent plots with the same category, and the other one based on categories, where the plots are grouped into categories. For the compactness based on patches the function is defined as:

$$Compactness = 4\pi \sum_{i=1}^{C} w_i \left(\frac{\sum_{j=1}^{NP_i} \frac{A_{ij}}{P_{ij}^2}}{NP_i} \right) \qquad (3)$$

where NP_i is the number of patches of the *i-th* category, A_{ij} and P_{ij} are the area and perimeter of the *j-th* patch of the *i-th* category, respectively. This formula is based on the fact that, for a given area value, the so called circularity is maximized by a circle (and the maximum is 1) [19]. The compactness function based on categories:

$$Compactness = 4\pi \sum_{i=1}^{C} w_i \left(\frac{\sum_{j=1}^{N} a_{ij}}{\sum_{j=1}^{N} p_{ij}^2} \right) \qquad (4)$$

where p_{ij} is the perimeter of the *j-th* plot allocated to the *i-th* category. Note that this function has clearly a lower computational cost than (3) because it avoids the computation of patches.

4.2 Computational issues

Computing the objective function for a trial solution E_t is done by calculating the variation of E due to the change of the category of the involved plot instead of calculating the overall suitability and compactness for the whole plot map. The new value of the suitability subobjective is calculated by subtracting the area-weighted suitability of the changed plot for the old category and by adding the area-weighted suitability of the changed plot for the new category.

In the case of the compactness function based on patches, the compactness score is computed from the area and perimeter of each patch. In the calculation of the compactness of the new category three situations can be distinguished; i) a new patch is generated, ii) the area of an existing patch increases, or iii) several patches are merged. In the calculation of the compactness of the old category also other three situations can happen; i) a patch disappears, ii) the area of an existing patch decreases, or iii) a patch is divided into several patches. The identification and management of these situations is performed as follows. In the case of the new category, if no neighbour plot has the new category, a new patch is generated that has the area and perimeter of the changed plot. If a neighbour plot whose category is the same as the new one is found, the patch is reconstructed from the changed plot using the neighbour patch ID. This case can correspond to any of the following two situations: the area of an existing patch simply has increased or several patches have been merged. In the latter case other neighbour plots with the new category and different patch ID will be found, so this patch ID is removed since that patch has been merged with the previous reconstructed patch. In the case of the old category, it is considered that the patch has disappeared unless a neighbour plot whose category coincides with the old category is found. From this neighbour plot, a patch is reconstructed by a recursive flooding algorithm and the plots that constitute the reconstructed patch are identified. If other neighbour plot that has the old category, and not included in the reconstructed patch is found, a new patch is generated from this neighbour plot because the old patch has been fragmented.

The compactness function based on patches presents an interesting issue when several patches are merged. Since the overall compactness is the average of the compactness of each patch, when two patches with relatively high compactness are merged, the resulting patch often has lower compactness and consequently generates lower overall compactness. It is important to deal with these situations because merging patches produces benefits at long term. Therefore a mechanism to promote the conservation of changes that merge patches has been introduced. This mechanism consists on increasing the temperature of the Boltzmann test by a certain factor when the number of patches of the new solution is lower than the number of patches of the old solution in order to increase the probability of acceptance of the new solution. A multiplying factor is introduced and tuned by the planner to control this mechanism.

In the case of the compactness function based on categories, the new value of this subobjective is calculated by modifying the area and the perimeter of the old and new categories. The area is modified by subtracting the area of the changed plot from the compactness score of the old category and by adding it to the compactness score of the new category. The new values of perimeter for each category are obtained by comparing the old and new categories of the changed plot with the category of their neighbour plots.

4.3 The annealing schedule

The parameters of the annealing schedule must be defined by the planner. In general, it is recommended that the initial value of T ensure that about 80% of trials are successful at this stage; this value will depend on both the way in which the objective function varies with configuration, and the configuration generating scheme, and must be identified by trial and error for each problem. The heat balance condition, that is, the number of iterations executed at each temperature, was approximately twice the number of plots and each reduction of T was affected by multiplying it by a constant factor, which was 0.95 by default. The stop condition of the algorithm is the number of temperatures established by the planner, which was set to 200 by default.

Figure 2 shows the evolution of compactness and suitability with the temperature using the compactness function based on categories. The compactness function based on patches has similar behaviour. In shown case the compactness values and the variations of these values are very low, so a higher weight must be assigned to the compactness subobjective.

In Figure 3 note that the compactness function based on patches tends to generate a greater number of patches. In this figure each category is identified with a different grey level.

5. Parallel simulated annealing

The computational cost of the algorithm is high due to the large number of plots and the implicit nature of the problem. To get a more practical algorithm, the execution time has to be reduced, and the solution lies in its parallelization. Two strategies to parallelize the simulated annealing are proposed: parameter parallelization and spatial parallelization. A third possible strategy was also considered based on the parallelization of the computation of the objective function. However we found out that it is not efficient, mainly because the low computational cost of the objective function as, according to our proposal, only changes caused by the changes in the category of a single plot are taken into account.

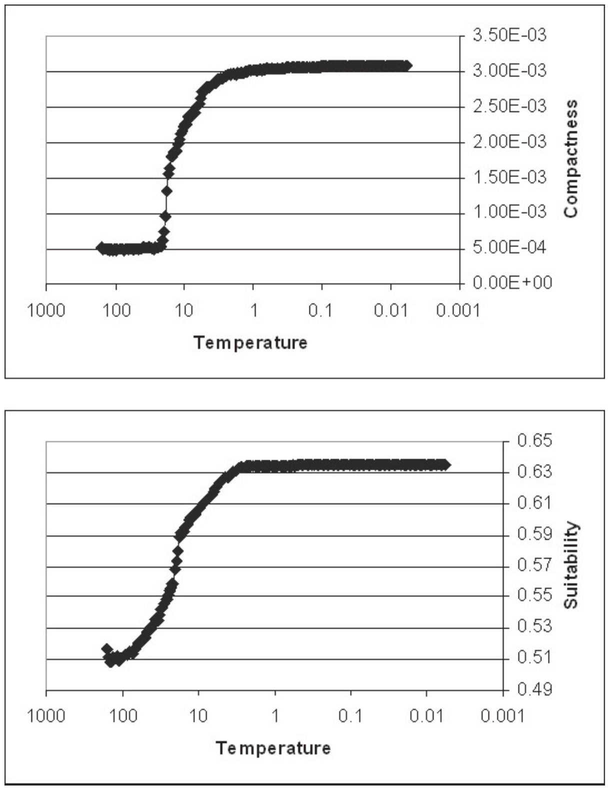

Fig. 2: Compactness and suitability evolution with the temperature for compactness function based on categories or patches.

5.1 Parameter parallelization

The identification of the optimal values of the parameters that guide the annealing is key issue to find the best solution. Running the algorithm in parallel with different parameter values helps in the search of the optimal values. These parameters are: the initial temperature, the number of iterations for each temperature, the cooling coefficient, the initial solution and the weights of both subobjectives. The parallelization uses as many processes as the number of different initial temperatures in this case of study.

5.2 Spatial parallelization

The spatial parallelization implies that each process run the algorithm in a particular geographic zone of the study area. The plot map is partitioned into groups of plots that are completely surrounded by plots allocated to the fixed categories, that is, by plots excluded from the simulation. In this way, there are no borderline interactions among zones. Each of these isolated groups of plots is called a cluster. The algorithm identifies the clusters, from the plot map by a

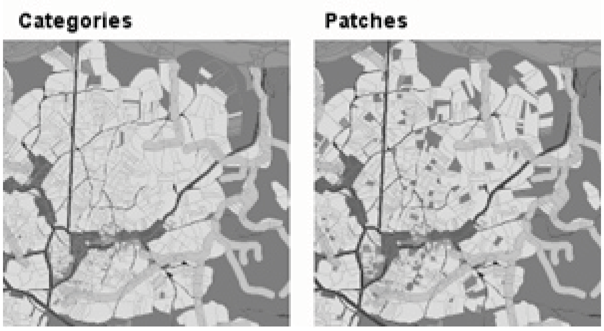

Fig. 3: Land zoning maps obtained with both objective functions; the first one uses the compactness function based on categories and the second one the compactness function based on patches.

pre-processing stage, using a flooding algorithm. In order to balance the computational load, the clusters are distributed among the processes so that the number of plots in each process is as similar as possible to the others.

The execution of each process is practically independent from the rest of them. The only common data accessed by all the processes is the value of the total area allocated to each category. Note that this area is constrained between a certain minimum and a maximum. Therefore changes of the category of a plot that results in a total area for a category exceeding the minimum and maximum values cannot be done. Therefore this constraint must be checked continuously using mutual exclusion operations.

6. Case of study

As a case study, a part of a Galician municipality called Guitiriz, was considered, it consists of 36,803 plots. After the pre-processing stage 34,000 polygons do not have a fixed category, so it must be taken into account in the simulated annealing stage. The plot suitability for each category and the total area to be allocated to each category were obtained from previous studies [20]. All performance tests were executed in a system with 2 processors Intel Xeon E5440 2.83GHz, 4 cores each and 16GBs of shared memory.

6.1 Parameter parallelization

Figure 4 shows compactness and suitability dependence with the initial temperature and with different weights for both subobjectives labelled by the values of wc and ws respectively.

There is not a clear trend in the influence of initial temperature on objective function values, so the evaluation of a wide range of temperature values is important to find the best value of this parameter. Figure 5 shows the results obtained by running the algorithm three times with the same parameters but with different initial solutions. Differences between the solutions obtained with different initializations but the same parameters are similar to the differences obtained by varying the weighting of the subobjectives.

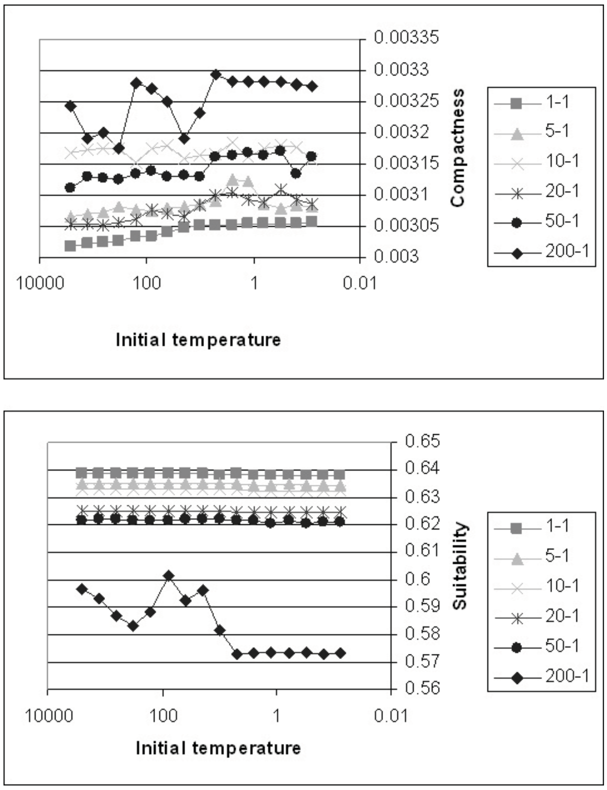

Fig. 4: Influence of the initial temperature on compactness and suitability by using the compactness function based on categories and different weights for the subobjectives.

6.2 Spatial parallelization

In figure 6 the influence of the initial temperature and the number of processes in the algorithm are shown. The increase in the number of processes decreases the suitability value but does not influence the compactness value. Note that influence on the initial temperature is very low.

Figure 7 shows the speedup for 7 different situations that are labelled as: the kind of algorithm (by patches or by categories), the values of w_c and w_s, and the initial temperature respectively.

The highest speedups are obtained with the compactness function based on patches and with the compactness function based on categories when only the compactness subobjective is optimized, because in these cases each process manages only local data.

6.3 The influence of the compactness

The compactness based on categories provides solutions with a number of patches quite lower than the compactness metric based on patches. Table 1 shows the number of patches in the final solution by using different compactness

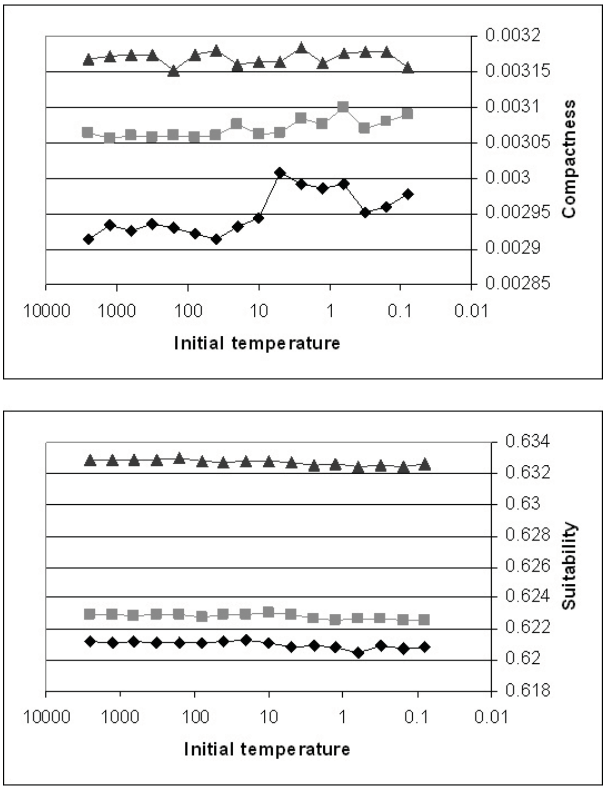

Fig. 5: Influence of the initial solution.

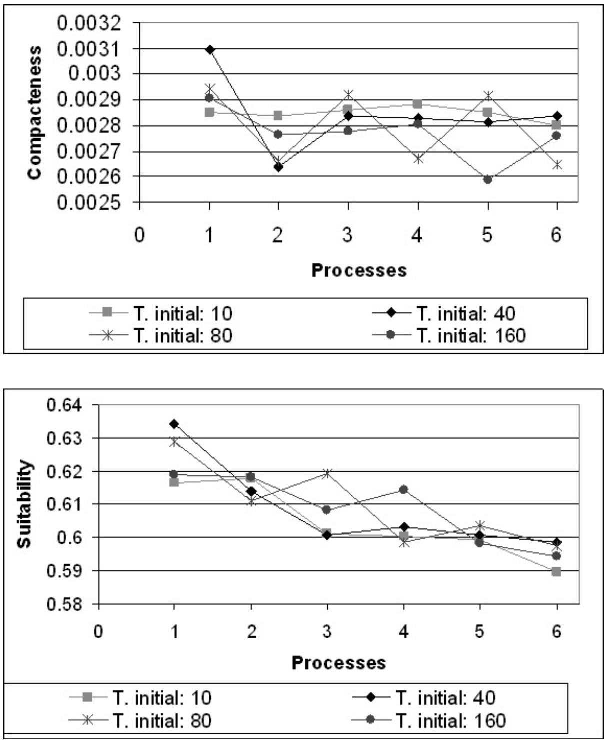

Fig. 6: Compactness and suitability values for each number of processes (the objective function uses the compactness function based on categories).

functions, different weights for subobjectives and different values for the temperature multiplier. Note that an increase on the value of the temperature multiplier factor reduces the number of patches. Anyway this number is higher than the number of patches generated by the compactness based on categories. The increase of the suitability subobjective weighting when using compactness function based on patches reduces the number of patches, because the spatial distribution of suitability presents a certain compactness by itself.

Table 1: Number of patches

Compactness	Wc	Ws	T multiplier	Patches number
Categories	0	1		7786
Categories	1	1		6361
Categories	5	1		5677
Categories	10	1		5496
Categories	20	1		5369
Categories	50	1		5235
Categories	1	0		4786
Patches	1	0	1	8842
Patches	1	0	100	7698
Patches	1	1	100	7193

6.4 Comparison with handmade planning

In order to evaluate the solutions provided by the algorithm, these solutions have been compared to the land use zoning map designed by technicians for the municipal land use plan of Guitiriz. The overlap area was measured as percentage of the total area of the technical solution. For compactness based on categories the coincidence was: agricultural 78%, forestry 68%, natural space 91% and urban 49%. For compactness based on patches the coincidence was: agricultural 78%, forestry 67%, natural space 87% and urban 49%. This results show a good matching for agricultural, forestry and natural space categories. The causes of the worst matching of urban category are the aesthetic and architectural criteria used by technicians in urban planning, which are not considered in the algorithm. However, the global suitability for the land zoning map designed by technicians is 0.52 and for the land zoning maps provided by the algorithm is around 0.62, so a significant improvement is achieved.

7. Conclusions

In this paper we deal with the problem of land use planning. Our proposal is to solve the delimitation of land use categories issue, that is frequently the stage of the

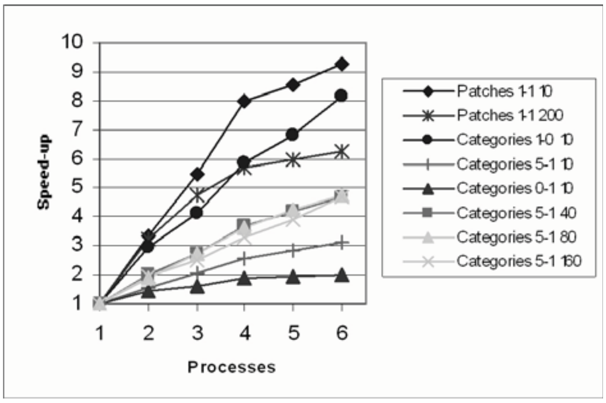

Fig. 7: Performance of the spatial Geographical parallelization.

whole process that is the bottleneck in practice. After a preprocessing stage, a simulated annealing based heuristic is used to efficiently solve the problem. An objective function that is a linear combination of two factors: the suitability and the compactness is introduced. The quality of the results on real situations are comparable to those obtained by experts. However the best parameters that rule the annealing can not be easily established prior to the execution. We used this feature to parallelize the algorithm by running the sequential code in several processes using different parameters. The benefits of this approach are mainly in the improvement on the quality of the final result. In addition, a spatial parallel implementation is proposed in which the geographical zone of study is partitioned into a number of so called clusters that can be processed in parallel. Appropriate mechanisms to share the information among the processes have been implemented. The efficiency of this second parallel implementation was validated in a real case of study. Both parallel proposals are orthogonal, so they can be applied simultaneously.

As a future work, one of the most immediate improvements is to study other kind of functions for the evaluation of the compactness criteria. The optimization of other spatial metrics such as connectivity is also interesting, especially for the case of the natural space category in order to design ecological networks.

8. Acknowledgment

This work is included in the project named "Geographical Information Systems for Urban Planning and Land Management using Optimization Techniques on Multicore Processors" with code 08SIN011291PR, and Consolidation of Competitive Research Groups (ref. 2010/06 and 2010/28), funded by the Galician Regional Government, Spain. And by Ministry of Education and Science of Spain under contract TIN 2007-67537-C03.

References

[1] J. R. Eastman, W. Jin, P. A. K. Kyem, and J. Toledano, "Raster procedures for multi-criteria/multi-objective decisions," *Photogrammetric Engineering and Remote Sensing (PE&RS)*, vol. 61, no. 5, pp. 539–547, May 1995.

[2] R. G. Cromley and D. M. Hanink, "Scale-independent land-use allocation modeling in raster gis," *Cartography and Geographic Information Science*, vol. 30, pp. 343–350(8), 1 October 2003.

[3] J. C. J. H. Aerts, E. Eisinger, G. B. M. Heuvelink, and T. J. Stewart, "Using linear integer programming for multi-site land-use allocation," *Geographical Analysis*, vol. 35, no. 2, pp. 148–169, 2003.

[4] D. Nalle, J. Arthur, and J. Sessions, "Designing compact and contiguous reserve networks with a hybrid heuristic algorithm," *Forest Science*, vol. 48, no. 1, pp. 59–68(10), February 2002.

[5] T. J. Stewart, R. Janssen, and M. van Herwijnen, "A genetic algorithm approach to multiobjective land use planning," *Comput. Oper. Res.*, vol. 31, pp. 2293–2313, December 2004.

[6] K. B. Matthews, S. Craw, and A. R. Sibbald, "Implementation of a spatial decision support system for rural land use planning: integrating gis and environmental models with search and optimisation algorithms," *Computers and Electronics in Agriculture*, vol. 23, pp. 9–26, 1999.

[7] T. Cay and F. Iscan, "Fuzzy expert system for land reallocation in land consolidation," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11 055 – 11 071, 2011.

[8] J. Porta, J. Parapar, G. L. Taboada, R. Doallo, F. F. Rivera, I. Santé, M. Suárez, M. Boullón, and R. Crecente, "A java-based parallel genetic algorithm for the land use planning problem," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2011*, Dublin, Ireland, July 12-16, July 12-16 2011, accepted.

[9] J. C. J. H. Aerts and G. B. M. Heuvelink, "Using simulated annealing for resource allocation," *International Journal of Geographical Information Science*, vol. 16, no. 6, pp. 571 – 587, 2002.

[10] M. Boyland, J. Nelson, and F. L. Bunnell, "Creating land allocation zones for forest management: a simulated annealing approach," *Canadian Journal of Forest Research*, vol. 34, no. 8, pp. 1669–1682, 2004.

[11] E. Martínez-Falero, I. Trueba, A. Cazorla, and J. L. Alier, "Optimization of spatial allocation of agricultural activities," *Journal of Agricultural Engineering Research*, vol. 69, no. 1, pp. 1 – 13, 1998.

[12] I. Santé-Riveira, R. Crecente-Maseda, and D. Miranda-Barros, "Gis-based planning support system for rural land-use allocation," *Computers and Electronics in Agriculture*, vol. 63, no. 2, pp. 257 – 273, 2008.

[13] S. K. Sharma and B. G. Lees, "A comparison of simulated annealing and gis based mola for solving the problem of multi-objective land use assessment and allocation," in *17th International Conference on Multiple Criteria Decision Analysis*, Whistler, Canada, August 2004, pp. 6 –11.

[14] E. Onbasoglu and L. Ozdamar, "Parallel simulated annealing algorithms in global optimization," *Journal of Global Optimization*, vol. 19, pp. 27–50, 2001, 10.1023/A:1008350810199.

[15] Z. J. Czech, *Parallel and Distributed Computing*. InTech, January 2010, ch. A Parallel Simulated Annealing Algorithm as a Tool for Fitness Landscapes Exploration, pp. 247–272.

[16] M. Davis and J. Aquino, "Jts topology suite technical specifications," Vivid Solutions, Tech. Rep., 2003. [Online]. Available: http://www.vividsolutions.com/JTS/bin/JTS Technical Specs.pdf

[17] V. Olaya, "Sextante programming guide," Sextante, Tech. Rep., 2011. [Online]. Available: http://www.sextantegis.com

[18] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[19] R. S. Montero and E. Bribiesca, "State of the art of compactness and circularity measures," *International Mathematical Forum*, vol. 4, no. 27, pp. 1305–1335, 2009.

[20] I. Santé, R. Crecente, M. Boullón, and D. Miranda, *Spatial Decision Support for Urban and Environmental Planning. A Collection of Case Studies*. Malaysia: Arah Publications, 2009, ch. Optimising land use allocation at municipal level by combining multicriteria evaluation and linear programming, pp. 33–60.