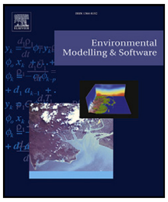
Environmental Modelling and Software 176 (2024) 106000

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



Position Paper

Spatially targeted afforestation to minimize sediment loss from a catchment: An efficient hill climbing method considering spatial interaction

Grethell Castillo-Reyes [a,b,*], René Estrella [c], Dirk Roose [b], Floris Abrams [d], Gerdys Jiménez-Moya [a], Jos Van Orshoven [d]

[a] Data Representation and Analysis Center, University of Informatic Sciences, Cuba
[b] Department of Computer Science, KU Leuven, Belgium
[c] Research Group Models, Analysis and Simulations, Department of Computer Science, University of Cuenca, Ecuador
[d] Division of Forest Nature and Landscape, Department of Earth and Environmental Sciences, KU Leuven, Belgium

ARTICLE INFO

Dataset link: https://gitlab.kuleuven.be/u0123674/acamf

Keywords:
Accelerated CAMF software
Afforestation
Hill climbing heuristic
Spatial optimization
Spatial interaction
Sediment loss

ABSTRACT

Based on soil erosion and sediment transport processes, CAMF (Cellular Automata-based heuristic for Minimizing Flow) selects sites for afforestation to minimize sediment influx at a catchment's outlet. CAMF uses a raster representation of the catchment and a steepest ascent hill-climbing optimization heuristic, safeguarding spatial interaction. Its execution time can be prohibitively long for large data-sets. Parallelization results in a speedup of 20 to 24 on 28 cores. We present variants of the optimization method to reduce the number and cost of the iterations. We present a tuning algorithm for the meta-parameters of these variants. The results obtained for two contrasting catchments illustrate that the accelerations reduce the cost by a factor larger than 100, with negligible effect on the afforested cells and magnitude of the sediment reduction. The results indicate that higher levels of spatial interaction have a stronger impact on the accuracy of the results and/or the execution time.

Software availability

Name of the software: A-CAMF 1.0

Developer: Grethell Castillo Reyes

Contact: Data Representation and Analysis Center, University of Informatic Sciences, San Antonio de los Baños Km 2$^1/_2$, Cuba; Department of Computer Science, KU Leuven, Celestijnenlaan 200A box 2402, 3001 Leuven, Belgium.

Year First Available: 2022

Hardware requirements: A multi-core processor.

Program language: C++

Software required: A-CAMF is implemented in C++ and can be executed on either Linux or Windows. The required libraries to run A-CAMF are: The Geographic Data Abstraction Library (GDAL) and OpenMP.

Program size: 187 KB

The source code is available at https://gitlab.kuleuven.be/u0123674/acamf

1. Introduction

Spatial optimization problems play an important role in land use planning and land management; for example, to locate the optimal sites to perform a certain intervention (Witlox, 2005; Nguyen et al., 2017; Strauch et al., 2019), aimed at maximizing financial and/or ecological benefits, expressed at the local (on-site) or global (off-site) level. Different types of spatial optimization problems arise, depending on the decision variables, the objective function(s), and constraints. Therefore, an appropriate optimization method should be selected for each problem type (Kaim et al., 2018). Spatial interaction, also called spatial inter-dependency, must be taken into account to solve such problems. Changing the state of a site may influence the state of other sites (Hayes and Wilson, 1971), so that the effect of an intervention at a site cannot be determined independently of the interventions in other sites (Gersmehl, 1970; Wang, 2017).

Some spatial optimization problems can be formulated as Linear Programming (LP) or Integer Programming (IP) problems, allowing to compute the global optimum using standard mathematical software, see e.g. Fischer and Church (2003), Karterakis et al. (2007), Orsi et al.

[*] Corresponding author at: Department of Computer Science, KU Leuven, Belgium.
E-mail addresses: gcreyes@uci.cu, grethell.castilloreyes@kuleuven.be (G. Castillo-Reyes), rene.estrella@ucuenca.edu.ec (R. Estrella), dirk.roose@kuleuven.be (D. Roose), floris.abrams@sckcen.be (F. Abrams), gejimenez@uci.cu (G. Jiménez-Moya), jos.vanorshoven@kuleuven.be (J. Van Orshoven).

https://doi.org/10.1016/j.envsoft.2024.106000
Received 3 October 2023; Received in revised form 20 February 2024; Accepted 27 February 2024
Available online 29 February 2024

(2011) and Sarma et al. (2015). The execution time depends on many factors, including the number of decision variables and constraints. LP and IP formulations have been extensively explored when the number of decision variables and constraints is small, as in Sarma et al. (2015) and Fischer and Church (2003). Even then, the execution time required to solve the LP or IP problem can be very long.

Hence, spatial optimization problems are typically solved by heuristic methods (Borges et al., 2002; Li, 2007; Maier et al., 2014). For example, Bettinger et al. (2003) present a comparison of eight heuristic planning techniques (including random search, simulated annealing, great deluge, threshold accepting, tabu search, and genetic algorithms) in terms of solution quality and the effort required to obtain a solution, illustrating the challenges of using heuristic methods for forest planning. Similar studies are presented in Heinonen and Pukkala (2004), Bachmatiuk et al. (2015) and Dong et al. (2015), where several heuristic techniques to solve harvest scheduling problems are compared. Shan et al. (2009) present general trends in using heuristics methods in spatial forest planning, suggesting a shift from exact analytical solution techniques to heuristics in this field.

For example, Genetic algorithms (GAs) are frequently used to optimize the application and combination of Best Management Practices (Kaini et al., 2012; Panagopoulos et al., 2013; Chichakly et al., 2013; Yang and Best, 2015). While GAs can be used for a variety of search and optimization problems, the successful exploration of large search spaces requires a large population size and many iterations, causing long execution times (Cibin and Chaubey, 2015). The performance of GAs can be sensitive to the values of the parameters used (mutation probability, …) (Arabi et al., 2006), and tuning these parameters can be expensive. Hence, spatial optimization problems solved by a GAs typically have a small set of decision variables. For large problems, special techniques are proposed, such as multi-level spatial optimization (Cibin and Chaubey, 2015).

This paper discusses a method and its corresponding software implementation, designed to identify the cells in a raster representation of a study area for an intervention meant to optimize an objective function that is subject to spatial interaction. The type of study area considered in this paper is a hydrological catchment, the intervention considered is afforestation and the objective is the minimization of the cumulative sediment loss, also called sediment yield and denoted as SY, from the catchment at its outlet, as a result of the afforestation of a predefined (constrained) number of to-be-identified cells.

The SY is determined by the local sediment production in the catchment and the sediment flow towards the outlet. Local sediment production refers to the amount of detached soil particles that are available at each site of the catchment to be transported to other sites. The sediment locally produced at a site depends, among other factors, on the land cover, and it is lower in afforested sites than in agriculture/pasture sites. Hence, afforesting a site ultimately reduces SY. Various models to simulate soil erosion, sediment production, and sediment transport are proposed in the literature, e.g. Wischmeier et al. (1965), Renard et al. (1991), Van Oost et al. (2000), Van Rompaey et al. (2001), Vanegas (2010), Kumar et al. (2022) and Domingues et al. (2020). These models vary in complexity and often contain parameters that must be calibrated with experimental data, determining the accuracy of the simulations.

The focus of this paper, however, does not lie on such models, but on the optimization method to solve the site-location problem, i.e. to select those sites that minimize SY when covered by forest, where the area to be afforested or the required reduction in SY are given as constraints. Domingues et al. (2020) mention ‘precision forest restoration’.

For the afforestation problem considered in this paper, an IP formulation is presented in Vanegas et al. (2009, 2012). The sediment accumulation in each raster cell is based on locally produced sediment, the Single Flow Direction (SFD) model (the D8 variant O’Callaghan and Mark, 1984), and a convex piecewise linear function modeling the sediment transport between cells. Solving this IP problem is only feasible for small data-sets (Vanegas et al., 2012). Therefore, Vanegas (2010) introduced a heuristic method, called Cellular Automata based heuristic for Minimizing Flow (CAMF).

In each iteration of CAMF, typically, only one candidate cell is selected for afforestation, namely the cell for which afforestation maximizes the reduction of SY. Hence, the optimization heuristic in CAMF is a steepest ascent hill climbing method (Michalewicz and Fogel, 2000). However, since afforesting a cell reduces SY, special techniques to escape local minima are not needed in this case, which simplifies the optimization process. Details are presented in Section 2, with emphasis on the convergence of the iterative process and the computational cost.

Vanegas et al. (2012) report on the application of CAMF to locate optimal sites for afforestation in a small catchment. Using the same local sediment production and sediment transport models as in the IP formulation, the afforested cells were identical to those obtained by solving the IP formulation using the Lingo software, while the execution time was shorter.

The CAMF method was further extended in several directions, including the introduction of Multiple Flow Direction (MFD) models in Estrella (2015) and Castillo-Reyes et al. (2023a). Note that in principle the IP formulation in Vanegas et al. (2009) could be adapted for a MFD model, but it is impractical since the number of constraints would be immense (Vanegas et al., 2012). In Castillo-Reyes et al. (2023a) we showed that the CAMF heuristic converges fast, i.e. SY decreases drastically as a function of the number of selected cells, since it selects in each iteration those cells for which the marginal contribution to the sediment yield reduction by afforestation is the highest. In addition, the method is robust since it is deterministic and its performance does not depend on algorithmic parameters.

We also analyzed the scalability of the method when the raster size increases. Especially when using the MFD model, the execution time per iteration grows faster than linearly with the raster size. This restricts the applicability of CAMF to mid-sized geo-databases.

Domingues et al. (2020) use a GA to solve a very similar problem as the one addressed in this paper, namely to locate raster cells for afforestation, to minimize soil erosion and deposition, simulated by the Unit Stream Power Based Erosion Deposition (USPED) model. A large population and many GA iterations are needed for a moderately large raster data-set, resulting in long execution times with no guarantee of finding the global optimum.

To be able to compute optimal sites for intervention in large geo-databases, we introduce in this paper adaptations to the steepest ascent hill climbing method used in CAMF, to reduce its execution time, while still considering the off-site impact. The latter depends on spatial interaction; thus, reflecting the real nature of sediment flow. The adaptations that we propose are based on the following hypothesis: (a) by selecting multiple cells per iteration, the number of iterations can substantially be reduced, and (b) by computing the effect on SY of tentatively afforesting each cell of only a subset of the candidate cells, the cost per iteration can be reduced. These adaptations affect the optimization result, but we show that they have only a limited effect on the solution quality. We also present a tuning algorithm to appropriately choose the values of the meta-parameters of the new algorithmic variants. We show that the execution time can further be reduced by executing the simulations, performed in each iteration, in parallel on a multi-core processor.

This paper is organized as follows: Section 2 summarizes the characteristics of CAMF and describes the computation of the sediment accumulation in a raster cell and presents the accelerations implemented to reduce the execution time. In Section 3 the data-sets for the experiments are described. The results of the experiments are presented in Section 4 and discussed in Section 5. Conclusions are given in Section 6.

2. The CAMF and A-CAMF methodology

2.1. The original CAMF method

The CAMF method, presented in Vanegas (2010) and subsequent papers, uses a raster geo-database describing several properties of the catchment. To apply it to the case of afforestation to minimize sediment loss, it requires a Digital Elevation Model (DEM), a land cover map, and the mean annual amount of sediment produced locally in each cell ($\mathrm{ton\,ha^{-1}\,yr^{-1}}$). Since afforestation reduces the sediment production, two values are required for each cell i: α_i^1 and α_i^2 denoting respectively the original sediment production and the production in case cell i would be afforested.

The sediment accumulated in cell i, denoted by SA_i, is the sum of the locally produced sediment and the sediment flowing into that cell from up-slope cells. The latter depends on the flow direction model (SFD, MFD-FD8 and MFD-D∞ are implemented in CAMF) and the sediment transport model. Several transport models are available in CAMF. In this paper, we use the model presented in Vanegas (2010) and also used in Estrella (2015) and Castillo-Reyes et al. (2023a): the amount of sediment leaving cell i is a convex piece-wise linear function of SA_i, depending on six parameters: retention capacity ρ_i^k, flow factor γ_i^k and saturation threshold σ_i^k, where $k = 1$ in case cell i is not afforested and $k = 2$ if afforested.

When a cell is afforested, the retention capacity and saturation threshold increase ($\rho_i^2 > \rho_i^1, \sigma_i^2 > \sigma_i^1$), and the local sediment production and flow factor decrease ($\alpha_i^2 < \alpha_i^1, \gamma_i^2 < \gamma_i^1$). As a result, afforesting a cell reduces the sediment in that cell and also the sediment that flows into the down-slope cells, and eventually into the outlet cell(s). Details of the computation of the sediment accumulation raster, further denoted as SA, can be found in Vanegas (2010), Castillo-Reyes et al. (2023a) and in Appendix A.

For optimization purposes, CAMF uses an iterative heuristic to select the cells that, when afforested, minimize the amount of sediment reaching the target cell(s), constrained by a predefined number of cells to be afforested or a target sediment yield reduction. Only a subset of the cells, which we designate as ‘candidate cells’ (typically the cells under specific land use types, e.g., agriculture and pasture lands) can be selected. In each iteration, each candidate cell is tentatively afforested (one by one) and the SA raster is recomputed. Hence the sediment yield SY, i.e., the sediment accumulated in the target cell(s), due to the afforestation of each candidate cell separately, is computed. The candidate cells are ranked according to the achieved SY and the cell(s) leading to the minimal SY, thus the maximal potential reduction of SY, denoted by SYR, are selected and marked as afforested. Typically, only one cell is selected in each iteration (Vanegas, 2010; Vanegas et al., 2012, 2014; Castillo-Reyes et al., 2023a).

Note that the CAMF method has been already adapted in different ways, including the selection of compact and contiguous sites in Vanegas et al. (2014) and Castillo-Reyes et al. (2023), and a simplification of the iterative process based on on-site criteria in Estrella et al. (2014b). In the latter variant, the results can differ strongly from those obtained with the original version of CAMF, since in that case spatial interaction is completely disregarded.

2.2. Optimization procedure in CAMF

We now present in detail the iterative process to select the cells where afforestation should take place to minimize the sediment yield SY at target cell(s), e.g. at the outlet, and thus to maximize the sediment yield reduction SYR.

The selected cells can be represented by a binary array $\mathbf{x}$ of length n, with n the number of candidate cells and $x_i = 1$ or 0 if the candidate cell i is afforested or not, respectively. $\mathbf{x}$ is the decision variable of the optimization problem. Initially $x_i = 0$ for each candidate cell i. The constrained optimization problem is to find the array $\mathbf{x}$ that minimizes $SY = F(\mathbf{x})$ where the function F is determined by the sediment production and transport models, constrained by e.g. the number of candidate cells that should be afforested or by a given reduction in SY. Starting with $\mathbf{x} = \mathbf{0}$, $\mathbf{x}$ is iteratively updated to minimize $SY = F(\mathbf{x})$. Assume that the value of $\mathbf{x}$ at the beginning of iteration step k is $\mathbf{x}^{k-1}$. By tentatively afforesting every candidate cell i, the corresponding sediment yield reduction $\Delta SY_i^k = SY(\mathbf{x}^{k-1}) - SY(\mathbf{x}^{k-1} + e_i)$ is computed, with $\mathbf{e}_i$ the ith unit vector, which represents the discrete analog of the partial derivative of SY w.r.t. x_i in $\mathbf{x}^{k-1}$. $\Delta SY_i^k > 0$ for each cell i, since afforestation of a cell reduces SY. If $\Delta SY_j^k = \max_i \Delta SY_i^k$, cell j is selected in iteration k, and $\mathbf{x}^k = \mathbf{x}^{k-1} + \mathbf{e}_j$, i.e., a step in the jth direction in the decision variable $\mathbf{x}$ is taken.

This iterative heuristic is a steepest ascent hill climbing method to solve the discrete binary constrained optimization problem. When a convex transport function is used, as in this paper, CAMF will converge to the global optimum of the convex problem (Vanegas et al., 2012).

Due to spatial interaction, the effect of afforesting a particular cell on the achieved SY can vary in subsequent iterations. Therefore, if the flow paths to the target cell(s) of selected cell j and a not yet afforested cell i partly coincide, then in the next iteration, $SY(\mathbf{x}^k + \mathbf{e}_i)$ must be computed, requiring the computation of the sediment accumulation SA_p for all cells p on these flow paths; otherwise $SY(\mathbf{x}^k + \mathbf{e}_i) = SY(\mathbf{x}^{k-1} + \mathbf{e}_i)$. Checking whether j and i have partly coinciding flow paths is simple when the SFD method is used, but it is prohibitively expensive when MFD is used. Therefore, in our implementation, every not yet afforested candidate cell is tentatively afforested in each iteration.

Castillo-Reyes et al. (2023a) reported on using CAMF for dealing with the afforestation problem in two river catchments with different properties, while analyzing the scalability of this method for different problem sizes. To fully take into account spatial interaction, in each iteration k, $SY(\mathbf{x}^{k-1} + \mathbf{e}_i)$ and thus the sediment accumulation matrix SA must be computed for a large number of candidate cells. As discussed in Castillo-Reyes et al. (2023a), the latter requires traversing a large fraction of the cells in case a MFD model is used. Hence, in the worst case, the computational cost of one iteration is quadratic in function of the number of raster cells, since the number of the candidate cells can be nearly equal to the total number of cells. In addition, typically, each iteration selects only one cell, hence many iterations are executed to solve the constrained optimization problem.

However, in case the impact of spatial interaction is rather weak, the optimization heuristic described above could be relaxed. Indeed, Table 1 shows, for 10 iterations of CAMF for a real data-set, the 10 best-ranked candidate cells, according to the potential SYR that would be achieved by afforesting each cell separately. These highest-ranked cells are identified in iteration 1, and there is no need to iteratively rank them.

Let SY^0 be the original sediment yield (before starting the iterations); SY_i^k the sediment yield if cell i would be afforested in iteration k (taking into account the afforestation of cells in iterations $1, \ldots, k-1$); SYR_i^k the total sediment yield reduction if cell i would be afforested in iteration k,

$$SYR_i^k = SY^0 - SY_i^k \tag{1}$$

Note that in Table 1, SYR_i^k includes the reduction due to the afforestation of cells in previous iterations, but not the reduction due to the (concurrent) afforestation of other cells in the same iteration. In the later iterations, the SYR_i^k values do not differ much. The last column shows the cells selected in these first 10 iterations of CAMF. The cell selected in iteration k was at rank k in iteration 1, for $k = 1, \ldots, 10$. Hence, in this case, there is no difference between executing 10 CAMF iterations, with one cell selected per iteration, and selecting the 10 best-ranked cells in one iteration.

Although Table 1 presents an ideal situation w.r.t. the ranking of candidate cells in subsequent iterations, the above mentioned observations form the basis for the adaptations of the optimization heuristic presented in the next subsection.

Table 1
Top-10 of the candidate cells in the first iterations of an experiment with the original CAMF using a real data-set and the final selected cells to be afforested. x, y represent the raster coordinates of the cell; SYR_i^k is the sediment yield reduction if cell at x, y would be afforested in iteration k.

k	Iteration 1		Iteration 2		Iteration 3		…	Iteration 10		Selected cells	
	x, y	SYR_i^1	x, y	SYR_i^2	x, y	SYR_i^3		x, y	SYR_i^{10}	x, y	SYR
1	180,113	93.15	145,194	183.12	169,105	271.50		174,119	728.22	180,113	93.15
2	145,194	89.97	169,105	181.53	180,116	254.64		303,96	727.91	145,194	183.12
3	169,105	88.38	180,116	164.67	170,104	249.99		174,109	727.38	169,105	271.50
4	180,116	71.52	170,104	160.02	169,104	248.68	⋮	100,246	727.18	180,116	343.02
5	170,104	66.86	169,104	158.71	94,160	248.29		251,43	726.58	170,104	409.89
6	169,104	65.55	94,160	158.32	180,114	246.29		249,39	725.22	169,104	475.45
7	94,160	65.16	180,114	156.32	286,25	245.38		94,162	725.11	94,160	540.62
8	180,114	63.17	286,25	155.41	174,119	245.29		171,104	724.58	180,114	603.79
9	286,25	62.25	174,119	155.32	303,96	244.98		151,41	722.97	286,25	666.05
10	174,119	62.17	303,96	155.01	174,109	244.45		175,109	722.49	174,119	728.22
…	…	…	…	…	…	…	…	…	…	…	…

2.3. Accelerations implemented in A-CAMF

We now describe the adaptations of the optimization heuristic implemented in A-CAMF, the accelerated version of the CAMF method and software.

2.3.1. Selecting several cells per iteration

If in iteration k several cells i have nearly the same potential for sediment yield reduction, several cells can be selected in one iteration, reducing the number of iterations, while the impact on the accuracy of the results depends on the strength of spatial interaction in the region. Indeed, due to spatial interaction, the set of selected cells will not always be equal. The *SYR* obtained with A-CAMF can be smaller than the *SYR* obtained with CAMF, and this difference in *SYR* should be controlled.

A user-defined relative threshold T is used to determine how many cells are selected in an iteration, as follows. The *SY* and the *SYR* obtained at the end of iteration k are denoted by respectively SY^k and SYR^k; the *SY* and the *SYR* obtained in iteration $k+1$ by (tentatively) afforesting cell c are denoted by SY_c^{k+1} and SYR_c^{k+1}; the extra *SYR*, obtained in iteration $k+1$ by (tentatively) afforesting cell c, is given by

$$\Delta SYR_c^{k+1} = SYR_c^{k+1} - SYR^k = SY^k - SY_c^{k+1} \tag{2}$$

Let cell i be the highest-ranked cell in iteration $k+1$. The cells j for which the relative difference is smaller than threshold T, i.e.

$$RD_{ij} = \frac{\Delta SYR_i^{k+1} - \Delta SYR_j^{k+1}}{\Delta SYR_i^{k+1}} \leq T \tag{3}$$

are selected in iteration k in A-CAMF.

2.3.2. Partial ranking

Building the ranking, by simulating the sediment flow for each candidate cell, is computationally expensive. In Table 1, the top of the ranking in subsequent iterations consists of the same cells, sometimes in a different ordering, minus the cells that were selected in previous iterations. A similar behavior is observed in subsequent iterations. Also, the cells that were ranked low in iteration k are not generally ranked high in subsequent iterations. Thus, computing the effect of afforesting these low-ranked cells can be avoided.

Hence, to further accelerate the procedure, the complete ranking is only calculated in iterations $1 + r \times K$, $r = 0, 1, 2, \ldots$. In the intermediate K iterations, we only rank the 'top-N' cells of the ranking produced in the last iteration in which the complete ranking was computed. Hence, in these K iterations, only a subset of the candidate cells are ranked. We call this acceleration method 'partial ranking'. Since K determines how often the complete ranking is computed and N represents the number of cells that are ranked in the intermediate K iterations, decreasing N and/or increasing K reduces the computational cost, but this can increase the difference between the solution sets of CAMF and A-CAMF and thus decrease *SYR*. When $T > 0$ and the partial ranking is used, the number of cells selected in every iteration is limited to N.

2.3.3. Parallelization

The adaptations presented in Sections 2.3.1 and 2.3.2 aim at improving the computational cost of CAMF by reducing the number of iterations and the cost of these iterations, but still many simulations with high computational costs are required.

In each iteration of CAMF, the sediment flow is simulated for each (tentatively) afforested candidate cell. Since these simulations are independent of each other, they can be executed in parallel on a multi-core computer, resulting in a reduction of the wall clock time.

For the implementation Open Multi-Processing (OpenMP) directives (Chapman et al., 2007) are used in A-CAMF. The *omp loop schedule* directive indicates to the compiler how the loop iterations must be distributed among the available threads (cores). Since the simulations for each candidate cell can have different execution times, it is important to use dynamic scheduling, with a predefined size of the subsets of candidate cells (chunks) handled together to minimize the load imbalance. This ensures that no core remains idle while there are still tasks to be executed (Fig. 1 and Algorithm 1).

Algorithm 1 shows the pseudo-code of the accelerations included in A-CAMF.

2.3.4. Hyper-parameter tuning

We aim at setting the values of the acceleration parameters T (see Section 2.3.1) and K, N (see Section 2.3.2), such that the execution time is maximally reduced, while the relative difference (RD_n), in SYR values when afforesting n cells with the original CAMF and A-CAMF is smaller than a user-defined value (RD_{max}).

In the tuning process, we afforest m cells with CAMF and A-CAMF, with $m \ll n$, varying the acceleration parameter values such that $RD_m \leq RD_{max}$. Although this does not guarantee that $RD_n \leq RD_{max}$, the results in Section 4 indicate that this holds in general. m is set by the user.

The acceleration parameters are varied using a guided grid search. Indeed, we aim at a compromise between the computational cost and the accuracy of the results, which is different from the aim of classical methods to find parameter values for which the best results are obtained, e.g., grid search (Kim, 1997; Jiménez et al., 2007). For large values of the relative threshold T and the number of intermediate iterations K, the execution time of A-CAMF is strongly reduced, while for $T = 0$, $K = 0$, A-CAMF is identical to CAMF ($RD_m = 0$).

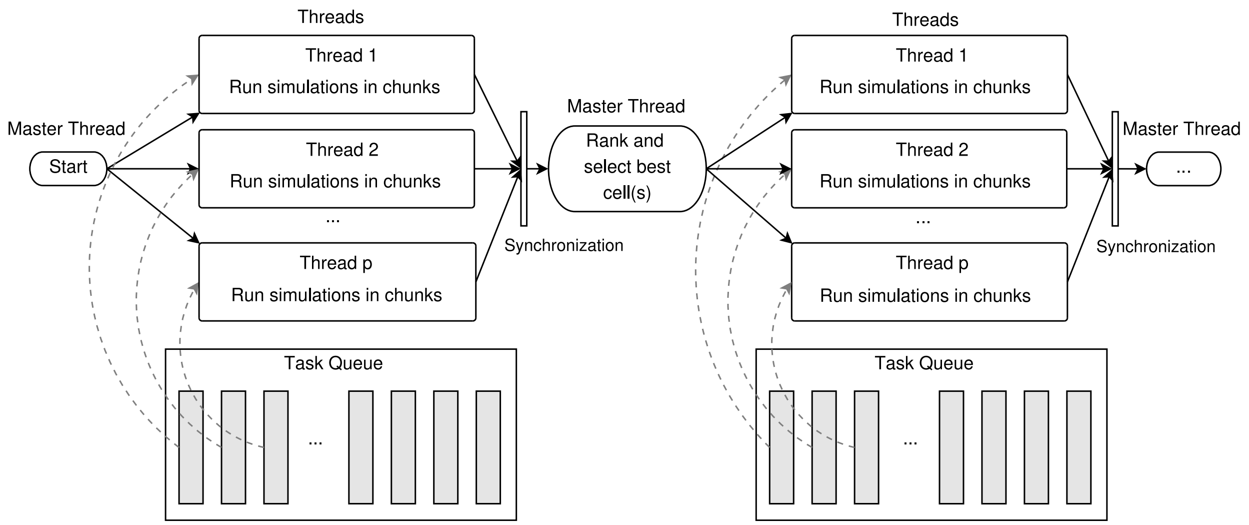

Threads
Thread 1
Run simulations in chunks
Thread 2
Run simulations in chunks
...
Thread p
Run simulations in chunks
Master Thread
Start
Synchronization
Master Thread
Rank and select best cell(s)
Task Queue
...
Threads
Thread 1
Run simulations in chunks
Thread 2
Run simulations in chunks
...
Thread p
Run simulations in chunks
Master Thread
...
Synchronization
Task Queue
...

Fig. 1. Representation of the parallel execution of the first iterations, using p threads and dynamic load balancing in A-CAMF.

Algorithm 1 Determine the cells to be selected for afforestation with A-CAMF

```
Input: Number of cells to be selected $n$, threshold for selecting multiple cells per iteration $T$, interval to compute the complete ranking $K$, size of partial ranking $N$

1. Create data structure $S$ to store cells to be selected for afforestation
2. $k \leftarrow 1$
3. $r \leftarrow 0$

while size of $S < n$ do
    if $k = 1 + (r \times K)$ then
        Start parallelism — distribute iterations by means of the dynamic schedule strategy
        for each candidate cell $i$ do
            4. Compute $SA_i$ raster and $SYR_i^k$ by tentatively afforesting cell $i$
        end for
        End parallelism
        5. Rank and select best cells by executing Algorithm 2 with $N = n$
    else
        Start parallelism — distribute iterations by means of the dynamic schedule strategy
        for first $N$ cells in complete ranking do
            6. Compute $SA_i$ raster and $SYR_i^k$ by tentatively afforesting cell $i$
        end for
        End parallelism
        7. Rank 'top N' and select best cells by executing Algorithm 2
        8. $r \leftarrow r + 1$
    end if

    9. $k \leftarrow k + 1$
end while

Output: Set of cells selected for afforestation $S$
```

The grid search uses the following values for T and K:

$$S^T = [T_{max}, T_{max} - T_{step}, T_{max} - 2 \times T_{step}, \dots, 0]$$

$$S^K = [K_{max}, K_{max} - K_{step}, K_{max} - 2 \times K_{step}, \dots, 0]$$

Algorithm 2 Rank and select cells in iteration k

```
Input: Number of cells to be selected $n$, threshold for selecting multiple cells per iteration $T$, number of cells to be ranked $N$

1. Rank $N$ cells according to $SYR$
2. Put cell(s) with highest $SYR$ in solution set $S$    ▷ $S$ is a global variable used to store the cells to be selected

repeat    ▷ Starting from the first cell $j$ not selected in step 2
    3. Compute $RD_{ij}$ following Eq. (2) and (3) with $i$ the highest ranked cell in iteration $k$
    4. Put cell $j$ in solution set $S$
until $RD_{ij} > T$ or size of $S == n$
```

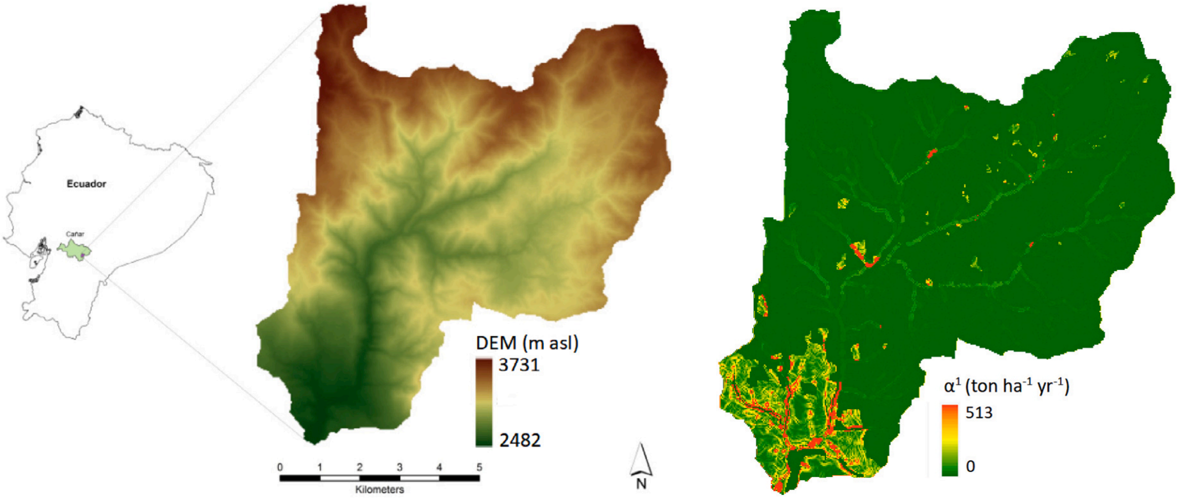
The tuning process consists of the following steps:

1. CAMF is executed to select m cells.
2. We first tune T, by running A-CAMF starting with $T = T_{max}$ and iterating through S^T, until a value for T is reached, for which $RD_m \leq RD_{max}$. In this step $K = 0$ and $N =$ number of candidate cells.
3. Once the value for T is selected, A-CAMF with partial ranking is executed starting with $K = K_{max}$, iterating through S^K. N is selected as follows. Suppose that in the last iteration of A-CAMF in step 2, R cells are selected, then in K iterations $\approx K \times R$ cells are selected. Since in the intermediate K iterations only N cells are ranked, the value of N should assure that all cells that will be afforested in K iterations are ranked. We set $N = K \times R$.

The algorithm ends when a set of parameters (T, K, N) is found, satisfying the condition $RD_m \leq RD_{max}$. The tuning algorithm is presented in Appendix B.

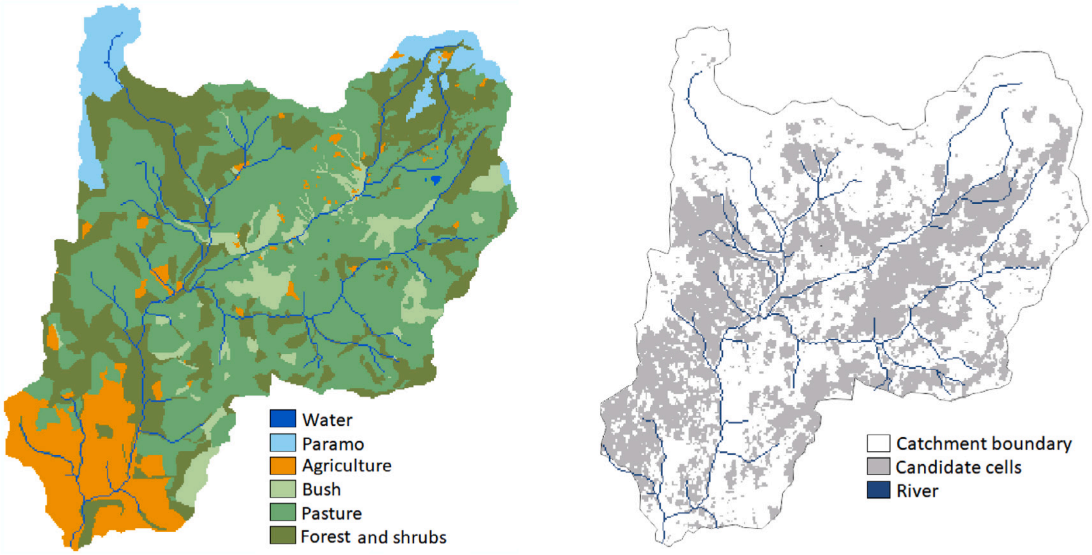
3. Study areas

To compare the performance of A-CAMF with the original CAMF, we used two raster geo-databases with different characteristics representing: (1) the Tabacay river catchment in Ecuador, and (2) the Maarkebeek river catchment in Belgium, see also Castillo-Reyes et al. (2023a).



(a) (b)

Fig. 2. Tabacay river catchment: (a) Location in Ecuador and Digital Elevation Model (Estrella et al., 2014a); (b) Initial local sediment production map α^1 (ton ha^{-1} yr^{-1}) calculated by means of RUSLE. $\min \alpha^1 = 0$ ton ha^{-1} yr^{-1}, $\max \alpha^1 = 513$ ton ha^{-1} yr^{-1}, mean $\alpha^1 = 2.54$ ton ha^{-1} yr^{-1}.



(a) (b)

Fig. 3. Tabacay river catchment: (a) Land cover map (Estrella, 2015); (b) Candidate cells (agricultural, pasture and shrubs cells) extracted from the land cover map.

To measure the difference between the set of selected cells by CAMF and A-CAMF, we used the Relative Spatial Coincidence (RSC), defined as

$$RSC = \frac{n-d}{n}, \tag{4}$$

with n being the number of selected cells and d being the number of cells that were not commonly selected.

3.1. Tabacay river catchment

The catchment is represented by 73 471 cells, called active cells, within a raster with 355 × 346 cells of 30 m × 30 m. The elevation ranges from 2 482 m to 3 731 m a.s.l., see the DEM in Fig. 2(a). Agricultural and pasture land cover ≈39% of the area, Fig. 3(a), leading to large amounts of sediment produced and transported to the outlet of the catchment, causing severe land degradation (Wijffels and Van Orshoven, 2009). A given subset of the agriculture, pasture, and shrubs cells are the candidate cells for afforestation (27 242 cells), see Fig. 3(b).

The initial sediment production map α^1 was computed using the Revised Universal Soil Loss Equation (RUSLE) (Renard et al., 1991), see Fig. 2(b). We refer to Castillo-Reyes et al. (2023a) for details about the computation and the parameters values used. For the other parameters in the SA calculation, i.e., retention capacities ρ^1, ρ^2; saturation thresholds σ^1, σ^2; flow factors γ^1, γ^2 and sediment production after afforestation α^2, listed in Appendix C, Table C.1, we used the values from Estrella (2015) and Castillo-Reyes et al. (2023a).

Additionally, we used two smaller data-sets obtained by clipping the original data-set around the outlet. Table 2 shows the growth factor of the number of active and candidate cells when the intermediate and original Tabacay data-sets are compared with the smallest data-set.

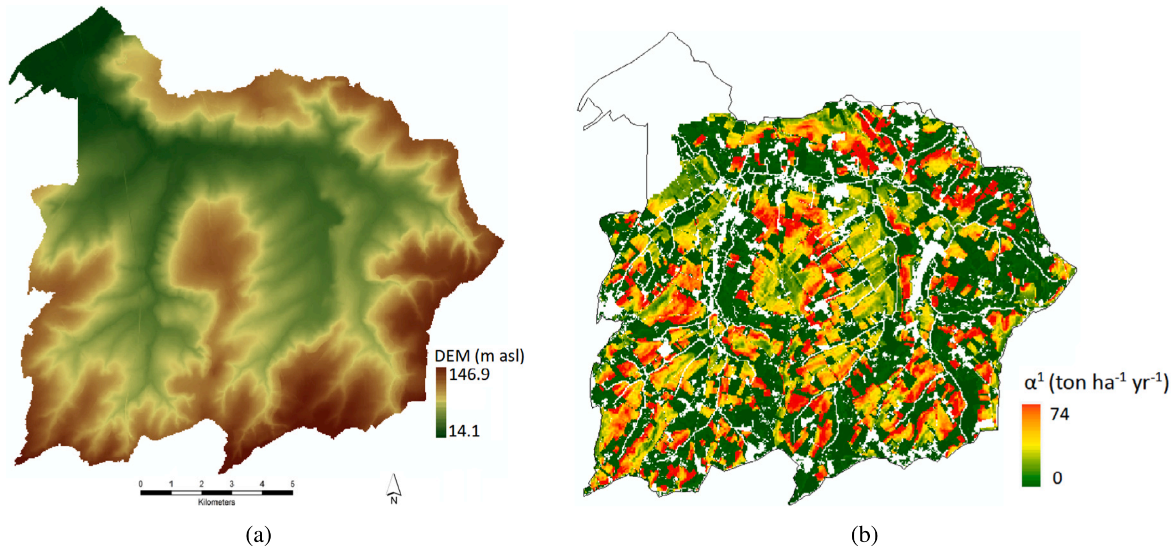


Fig. 4. Maarkebeek river catchment in Belgium: (a) Digital Elevation Model (Gabriels et al., 2022); (b) Initial local sediment production map α^1 ($\mathrm{ton\,ha^{-1}\,yr^{-1}}$) of the Maarkebeek river catchment calculated by means of RUSLE. $\min\alpha^1 = 0\ \ \mathrm{ton\,ha^{-1}\,yr^{-1}}$, $\max\alpha^1 = 74.38\ \ \mathrm{ton\,ha^{-1}\,yr^{-1}}$, mean $\alpha^1 = 1.91\ \ \mathrm{ton\,ha^{-1}\,yr^{-1}}$.

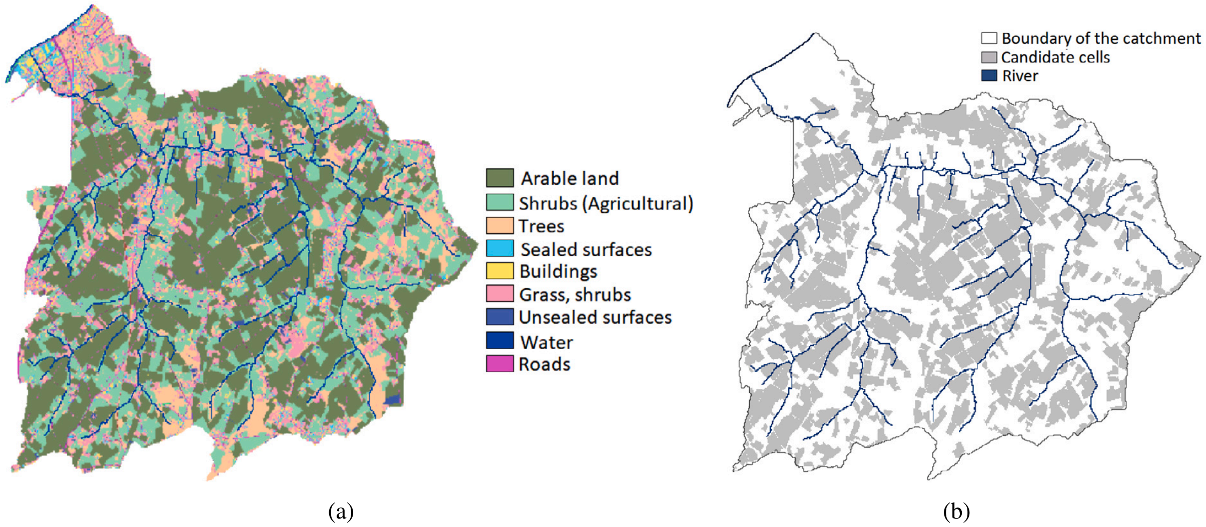


Fig. 5. Maarkebeek river catchment in Belgium: (a) Land cover map (Gabriels et al., 2022); (b) Candidate cells (agricultural, pasture, and shrubs cells) extracted from the land cover map.

Table 2
Number of active cells, candidate cells and growth factor in Tabacay data-sets. 'Small' and 'Intermediate' refers to data-sets corresponding to resp. $\frac{1}{16}$ and $\frac{1}{4}$ of the original data-set; 'Ratio 1': ratio of 'Intermediate' to 'Small'; 'Ratio 2': ratio of 'Original' to 'Small'; # active cells: number of raster cells covering the catchment; # candidate cells: number of cells than can be afforested.

	Small	Intermediate	Ratio 1	Original	Ratio 2
Dimensions	89 × 87	178 × 173		355 × 346	
# active cells	5475	22 494	4.11	73 471	13.42
# candidate cells	2259	9859	4.36	27 246	12.06

3.2. *Maarkebeek river catchment*

The Maarkebeek river catchment in Belgium is represented by 129 097 active cells within a raster with 464 × 438 cells of 20 m × 20 m. The elevation ranges from 14.1 m to 146.9 m a.s.l., see the DEM in Fig. 4(a). Agriculture covers a significant part of the area, while ≈10% is urbanized and ≈10% is afforested (Gabriels et al., 2022) (see the land cover map in Fig. 5(a)). The 53 792 candidate cells are shown in Fig. 5(b).

The initial local sediment production map α^1 (Fig. 4(b)) was also computed by RUSLE. We refer to Castillo-Reyes et al. (2023a) for more details about the computation of α^1, the parameters used for RUSLE and the parameters required for *SA* calculation. They are also in Appendix C, Table C.2.

4. Results

All experiments have been performed on a Xeon E5-2697 v3 CPU (2.6 GHz) with 28 cores and 128 GB of RAM, with Operating System Ubuntu Bionic Linux. In all cases, we use the MFD-FD8 variant of CAMF and A-CAMF.

4.1. *Speedup due to parallelization*

Table 3 shows the parallel execution times for one iteration on 1, 8, 16, and 28 cores when the small, intermediate, and the original

Table 3
Parallel CPU times (in seconds) for one CAMF iteration using 1, 8, 16 and 28 cores for the Tabacay data-sets.

# cores	Tabacay data-sets		
	Small	Intermediate	Original
1	1.37	26.67	651.20
8	0.18	3.52	96.24
16	0.10	1.95	51.67
28	0.06	1.12	30.22

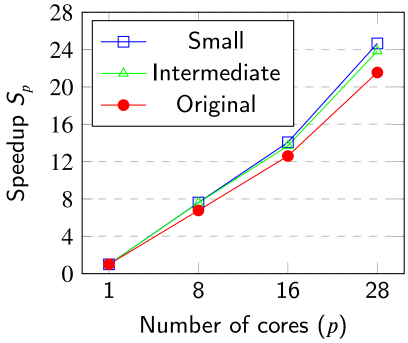


Fig. 6. Speedup S_p of CAMF (one iteration) using 1, 8, 16 and 28 cores for the Tabacay data-sets.

data-sets are used. The execution time increases more than linear with increasing number of active cells and candidate cells (Castillo-Reyes et al., 2023a). For the parallel code using p CPU cores, the speedup $S_p = \frac{T_s}{T_p}$, i.e., the ratio of the execution times of the sequential code T_s and the parallel code T_p, is shown in Fig. 6.

The 'ideal' speedup $S_p = p$ can be reached for completely parallelized algorithms. Fig. 1 and Algorithm 1 show that in A-CAMF, part of the operations are executed sequentially, e.g., the ranking computation. Also, synchronization among threads is required after each iteration. Hence, $S_p < p$. S_p is somewhat smaller for the original data-set than for the other data-sets, but in all cases S_p is close to the 'ideal' speedup p, due to the dynamic load balancing.

4.2. Speedup and accuracy due to the algorithmic adaptations

The selection of multiple cells per iteration and the use of a partial ranking, presented in Sections 2.3.1 and 2.3.2, aim at reducing the execution time, while the effect on *SYR* depends on the influence of spatial interaction. The latter can be controlled by tuning the acceleration parameters T, K and N. We evaluate the adaptations implemented using the two catchments presented in Section 3.

4.2.1. Tabacay catchment

For the original CAMF, the predicted *SY* and *SYR*, when 5%, 10%, 20% and 30% of the candidate cells are afforested in the Tabacay data-set, are shown in Table 4, indicating that afforestation of the first cells has the largest effect on *SYR*.

For A-CAMF, we first evaluate the effect of varying the threshold T with fixed values $K = 20$, $N = 1\,500$. Fig. 7(a) shows that increasing T reduces the number of iterations and thus the execution time, while the relative difference between *SYR* computed by CAMF and A-CAMF, denoted by *RD*, grows, but remains small ($RD < 0.4\%$). Indeed, as mentioned in Section 2, the set of selected cells in CAMF and A-CAMF can differ due to spatial interaction, leading to the observed RD.

Using even the smallest threshold ($T = 0.01$) reduces the number of iterations with a factor of 5 and 14 for afforesting respectively 5% and 30% of the candidate cells. Fig. 7(b) shows that, for a given T,

Table 4
SY and *SYR* computed by the original CAMF for afforesting 5%, 10%, 20% and 30% of the number of candidate cells of the original Tabacay data-set. SY^0: *SY* at initial situation; %*SYR*: $\frac{SYR \times 100}{SY^0}$.

CAMF	Afforested cells	*SY* (ton yr^{-1})	*SYR* (ton yr^{-1})	% *SYR*
$SY^0 = 36\ 774$ ton yr^{-1}	5% = 1362	25 530	11 244	31
	10% = 2724	23 829	12 945	33
	20% = 5448	22 739	14 034	36
	30% = 8172	22 463	14 310	36

Table 5
SY and *SYR* computed by the original CAMF for afforesting 5%, 10%, 20% and 30% of the number of candidate cells of the Maarkebeek data-set. SY^0: *SY* at initial situation; %*SYR*: $\frac{SYR \times 100}{SY^0}$.

CAMF	Afforested cells	*SY* (ton yr^{-1})	*SYR* (ton yr^{-1})	% *SYR*
$SY^0 = 30\,739$ ton yr^{-1}	5% = 2690	22 328	8 411	27
	10% = 5379	18 364	12 375	40
	20% = 10 758	16 239	14 500	47
	30% = 16 138	15 937	14 802	48

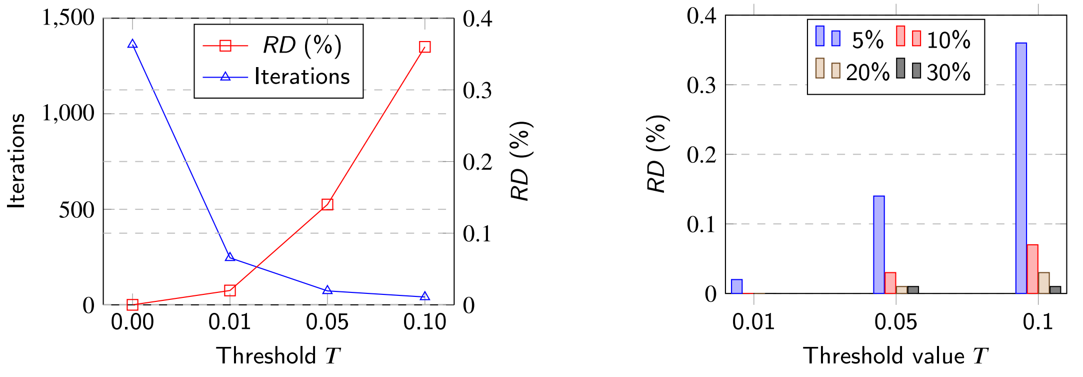
RD decreases with increasing number of afforested cells. When 30% of the candidate cells are selected, *RD* becomes negligible. This can be explained as follows. In the first iterations, the cells with the highest potential for *SYR* are selected; whether a cell is selected or not can cause a large *RD*. After many iterations, both CAMF and A-CAMF will have selected all cells with a high potential for *SYR*, and the selection of additional cells will not lead to a large *RD*.

The reduced number of iterations, due to threshold T, results in an algorithmic speedup, denoted by AS^T, which further increases with a factor proportional to K when partial ranking is used, since the cost of the intermediate iterations is much smaller than the cost of the iterations in which the whole ranking is computed. AS^T increases for a given T, with increasing number of cells to be afforested. Since a parallel speedup of ≈21 is achieved on 28 cores, the total speedup $TS^T_{28} \approx 21 \times AS^T$ is obtained by using the parallel A-CAMF instead of the sequential CAMF. Fig. 8(a) shows TS^T_{28} in function of the number of afforested cells, for several values of T while Fig. 8(b) presents TS^T_{28} as a function of T.

Fig. 9 shows the location of the selected cells, when 5% and 10% of the candidate cells are selected by CAMF and A-CAMF with $T = 0.1$, $K = 20$ and $N = 1\,500$. As the color codes indicate, A-CAMF selects almost the same cells as CAMF. The relative spatial coincidence *RSC*, as defined in Section 3, is presented in Fig. 10. In all considered cases $RSC > 99\%$.

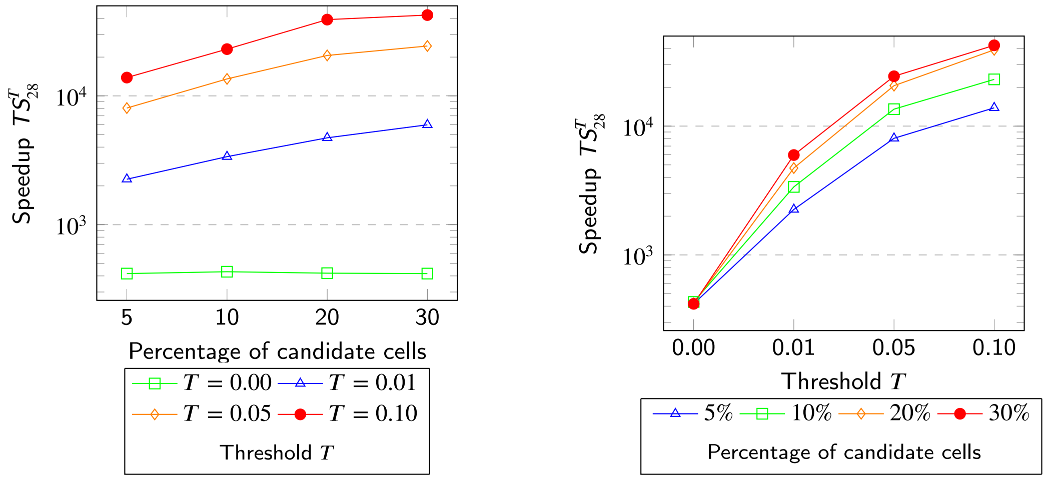
4.2.2. Maarkebeek catchment

Table 5 shows the SY and SYR computed by the original CAMF for afforesting 5%, 10%, 20% and 30% of the number of candidate cells of the Maarkebeek data-set. The results of executing A-CAMF with different values of T and $K = 10$, $N = 2\,000$ are presented in Figs. 11–14. Since the Maarkebeek data-set contains more cells than the Tabacay data-set, the execution times are substantially higher. When $T = 0$ and $K = 0$, selecting 30% of the candidate cells on 28 cores requires ≈ 26 days. The relative difference RD between the results of CAMF and A-CAMF (up to 2.8%) is much larger than for the Tabacay data-set for the same values of T, suggesting a stronger effect of spatial interaction in this region than in Tabacay. However, in this case, this difference also decreases when the number of selected cells increases, Fig. 11. The location of the selected cells and the RSC are presented respectively in Figs. 13 and 14, showing again the stronger effect of spatial interaction.



(a) (b)

Fig. 7. Performance of A-CAMF for varying threshold T: (a) Number of iterations and relative difference (RD) between SYR computed by CAMF and A-CAMF, for afforesting 5% of the candidate cells of the Tabacay data-set, with $K = 20$ and $N = 1\,500$; (b) Relative difference (RD) after afforesting 5%, 10%, 20% and 30% of the candidate cells of the Tabacay data-set, when A-CAMF is used with $K = 20$ and $N = 1\,500$ for different values of T.



(a) (b)

Fig. 8. Total speedup (TS_{28}^{T}): (a) for several values of T with respect to the percentage of candidate cells selected using the Tabacay data-set when multiple cells are selected per iteration and partial ranking is used with $K = 20$ and $N = 1\,500$; (b) for each set of selected cells (%) with respect to T using the Tabacay data-set when multiple cells are selected per iteration and partial ranking is used with $K = 20$ and $N = 1\,500$.

4.2.3. Hyper-parameter tuning process

We presented results obtained with A-CAMF with several values for T and fixed values for K and N, to evaluate how the accuracy and the execution time are affected. In practice, these parameters should be controlled by the tuning process presented in Section 2.3.4.

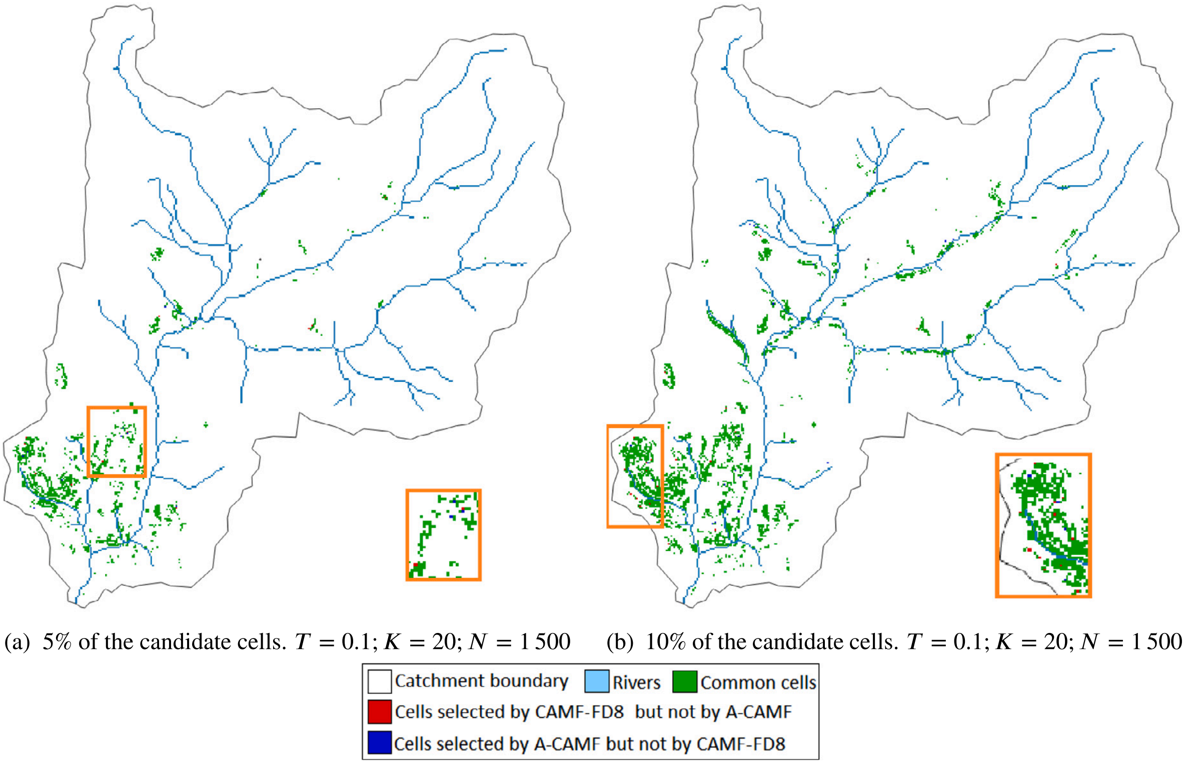
Table 6 shows the results of the hyper-parameter tuning process for the Tabacay data-set, where CAMF and A-CAMF are used to select m cells, where m is a small fraction (5% or 10%) of n – the number of cells that will eventually be selected – and n varies between 5% and 30% of the candidate cells. The selected values for the parameters T, K and N, obtained after the training iterations, ensure that for afforesting m cells the relative difference RD_m between the SYR values computed by CAMF and A-CAMF is less or equal to RD_{max}. Table 7 presents the performance of A-CAMF, executed with the tuned parameters T, K and N for $RD_{max} = 0.02\%$. When 20% and 30% of the cells are afforested, the relative difference for afforesting n cells, RD_n (Table 7) is nearly equal to RD_m (Table 6), indicating that the tune parameters are appropriately chosen. However, when 5% and 10% of the cells are afforested, the large difference between RD_n and RD_m indicates that, in this case, m, the number of cells used in the tuning process (68 and 136, respectively) is too small to tune the acceleration parameters. As in previous experiments, RD_n decreases when n, the number of selected cells, increases. Although 11 to 24 tuning iterations were carried out, the algorithmic speedup AS varies between 6 and 13, and the total speedup TS_{28} varies between 123 and 279. Note that T_{28}, AS and TS_{28} include the tuning processing time. In case A-CAMF is executed several times with similar input data, the tuning process must be executed only once.

5. Discussion

5.1. Analysis of the impact of spatial interaction

In Section 4.2 we compared the results of CAMF and A-CAMF for both data-sets, using RD, the relative difference in sediment yield reduction, and RSC, the relative spatial coincidence of the cells selected for afforestation. The adaptations implemented in A-CAMF have a different impact on the results for both regions. This difference is evident when A-CAMF selects n cells in only one iteration (by using a high threshold T), where n corresponds to 5% of the candidate cells. For the Tabacay catchment ($n = 1\,362$) $RD = 1.09\%$ and $RSC = 90.16\%$, while for the Maarkebeek catchment ($n = 2\,690$) $RD \approx 22\%$ and $RSC = 59.01\%$. This difference can be attributed to the influence of spatial interaction.

As observed by Castillo-Reyes et al. (2023a), the first cells selected by CAMF are typically concentrated in areas with high sediment production, as seen in the distribution of the selected cells in Fig. 9. A


Catchment boundary
Rivers
Common cells
Cells selected by CAMF-FD8 but not by A-CAMF
Cells selected by A-CAMF but not by CAMF-FD8

(a) 5% of the candidate cells. $T = 0.1; K = 20; N = 1\,500$ (b) 10% of the candidate cells. $T = 0.1; K = 20; N = 1\,500$

Fig. 9. Spatial coincidence of the afforested cells by CAMF and A-CAMF. Afforestation of (a) 5% and (b) 10% of the candidate cells in the Tabacay catchment, with $T = 0.1$, $K = 20$ and $N = 1\,500$. The color code indicates which cells are selected by both algorithms and which are selected by only one algorithm (visible when zooming in). The areas with larger differences are highlighted with orange rectangles.

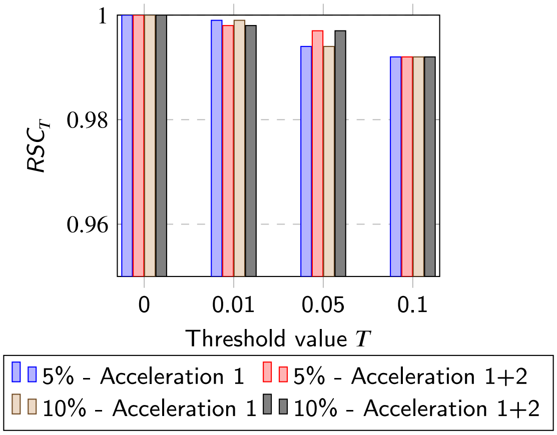


Fig. 10. Relative spatial coincidence (RSC) after afforesting 5% and 10% of the candidate cells of the Tabacay data-set, when A-CAMF is used with $K = 20$ and $N = 1\,500$ for different values of T. Acceleration 1: when multiple cells are selected per iteration; Acceleration 1+2: when additionally, partial ranking is used.

Table 6
Results of the hyper-parameter tuning process for the Tabacay catchment, with $RD_{max} = 0.02\%; T_{max} = 0.3; K_{max} = 50; T_{step} = 0.01; K_{step} = 5$. Afforested cells n: 5%, 10%, 20% and 30% of the candidate cells; m: fraction of the afforested cells used in the tuning process; T_{28}^{T}: CPU time (s) of the training process on 28 cores; RD_m: relative difference between the results of CAMF and A-CAMF (in %) $T; K; N$: parameter values after tuning.

# Afforested cells n	m	SYR (ton yr^{-1}) (CAMF)	# training iterations	T_{28}^{T}	SYR (ton yr^{-1}) (A-CAMF)	RD_m (%)	$T; K; N$
5% = 1362	5% = 68	3406.49	24	5 818.02	3406.08	0.01	0.17; 0; 0
	10% = 136	5214.95	11	6 677.84	5214.95	0.00	0.3; 0; 0
10% = 2724	5% = 136	5214.95	11	6 619.81	5214.95	0.00	0.3; 0; 0
	10% = 272	6980.42	11	12 787.49	6979.83	0.01	0.3; 5; 185
20% = 5448	5% = 272	6980.42	11	12 479.57	6979.83	0.01	0.3; 5; 185
	10% = 544	8675.75	11	24 392.72	8674.12	0.02	0.3; 0; 0
30% = 8172	5% = 408	7955.06	13	19 458.01	7954.95	0.00	0.28; 0; 0
	10% = 817	9782.24	21	42 424.60	9782.68	0.02	0.2; 0; 0

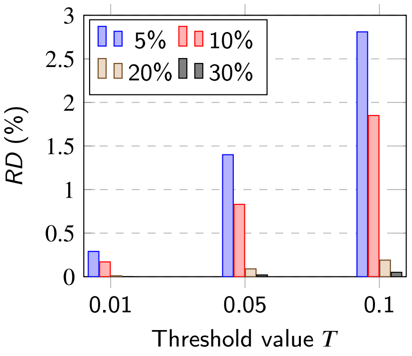
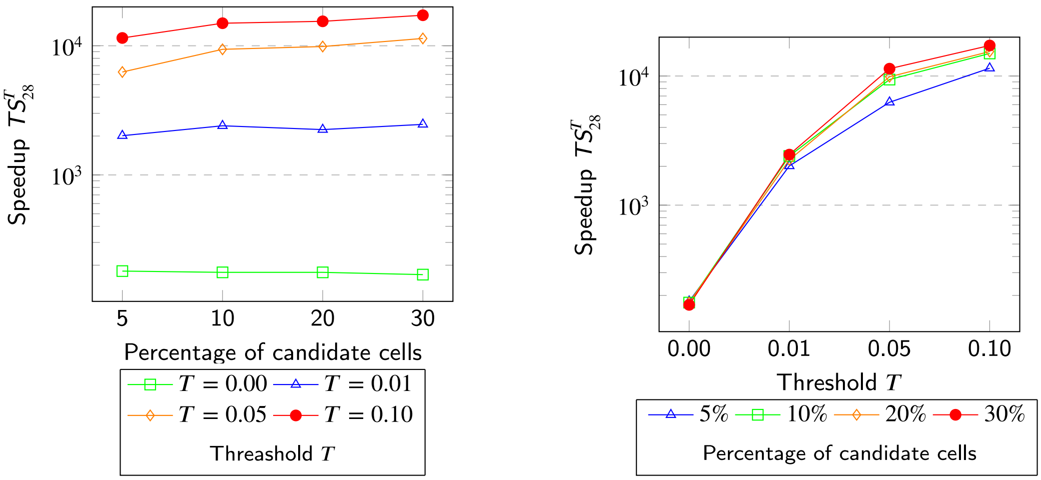


Fig. 11. Relative difference (RD) after afforesting 5%, 10%, 20% and 30% of the candidate cells of the Maarkebeek data-set, when A-CAMF is used with $K = 10$ and $N = 2\,000$ for different values of T.



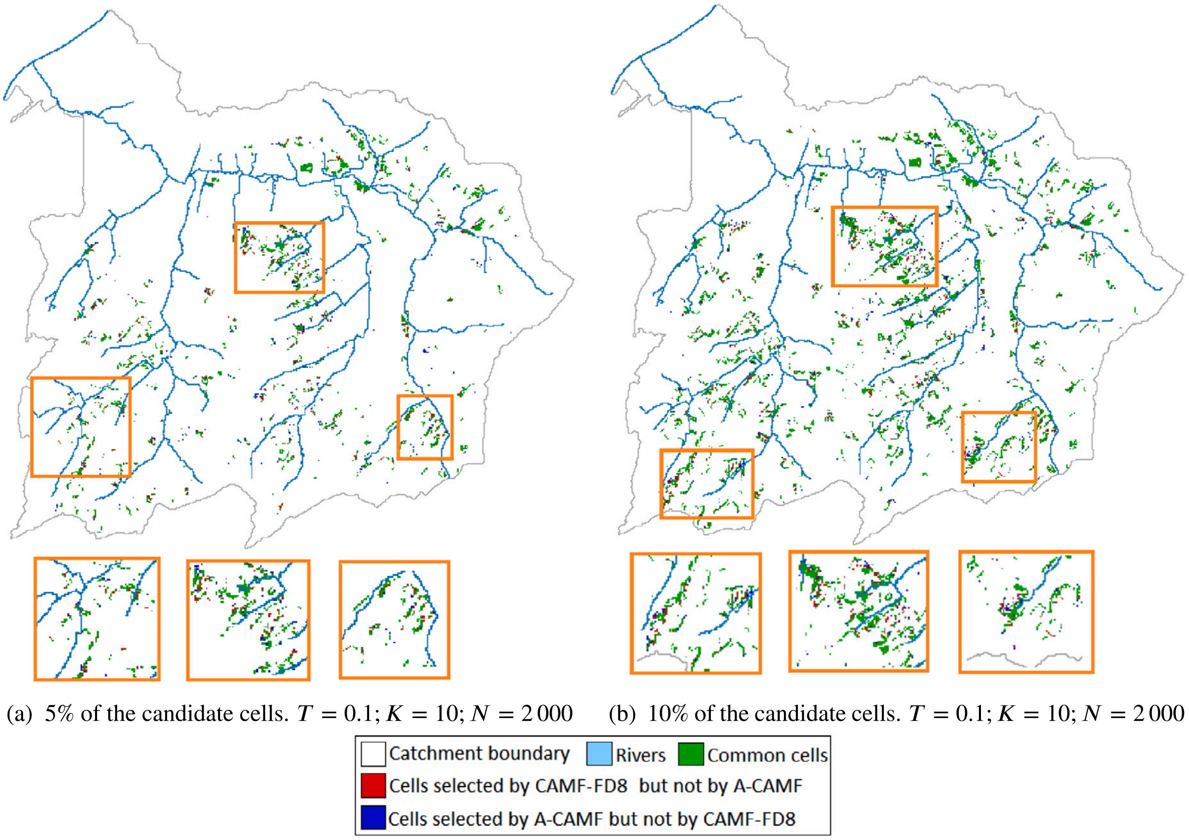
(a)

(b)

Fig. 12. Total speedup (TS_{28}^T): (a) for several values of T with respect to the percentage of candidate cells selected using the Maarkebeek data-set, when multiple cells are selected per iteration and partial ranking is used with $K = 10$ and $N = 2\,000$; (b) for each set of selected cells (%) with respect to T using the Maarkebeek data-set, when multiple cells are selected per iteration and partial ranking is used with $K = 10$ and $N = 2\,000$.

Table 7
Performance of A-CAMF when multiple cells are selected per iteration and partial ranking is used with the set of parameters $T; K; N$ selected in the hyper-parameter tuning process (Table 6) for the Tabacay catchment. Afforested cells n: 5%, 10%, 20% and 30% of the candidate cells; m: fraction of the afforested cells used in the tuning process; RD_n: relative difference between the results of CAMF and A-CAMF (in %); T_{28}^T: CPU time (s) on 28 cores; AS_{28}^T: Algorithmic Speedup; TS_{28}^T: Total Speedup.

# Afforested cells n	m	SYR (ton yr^{-1})	RD_n (%)	T_{28}^T	AS_{28}^T	TS_{28}^T
5% = 1362	5% = 68	11 170.23	0.65	6 475.49	6	137
	10% = 136	11 204.15	0.35	6 985.48	6	127
10% = 2724	5% = 136	12 907.09	0.29	7 005.01	12	253
	10% = 272	12 934.97	0.08	12 962.93	6	137
20% = 5448	5% = 272	14 031.61	0.02	12 729.62	13	279
	10% = 544	14 030.74	0.02	24 792.00	7	143
30% = 8172	5% = 408	14 306.94	0.02	20 016.97	12	166
	10% = 817	14 306.67	0.02	43 139.51	6	123



(a) 5% of the candidate cells. $T = 0.1$; $K = 10$; $N = 2\,000$

(b) 10% of the candidate cells. $T = 0.1$; $K = 10$; $N = 2\,000$

□ Catchment boundary
🟦 Rivers
🟩 Common cells
🟥 Cells selected by CAMF-FD8 but not by A-CAMF
🟦 Cells selected by A-CAMF but not by CAMF-FD8

Fig. 13. Spatial coincidence of the afforested cells by CAMF and A-CAMF. Afforestation of 5% (a) and 10% (b) of the candidate cells in the Maarkebeek catchment, when both algorithmic accelerations are active with $K = 10$ and $N = 2\,000$. The color code indicates which cells are selected by both algorithms and which are selected by only one algorithm (visible when zooming in). The areas with larger differences are highlighted with orange rectangles.

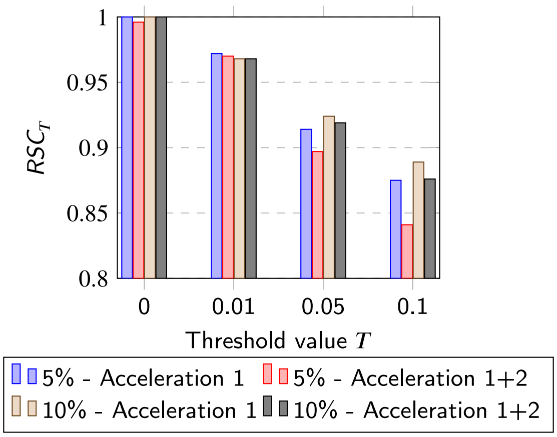
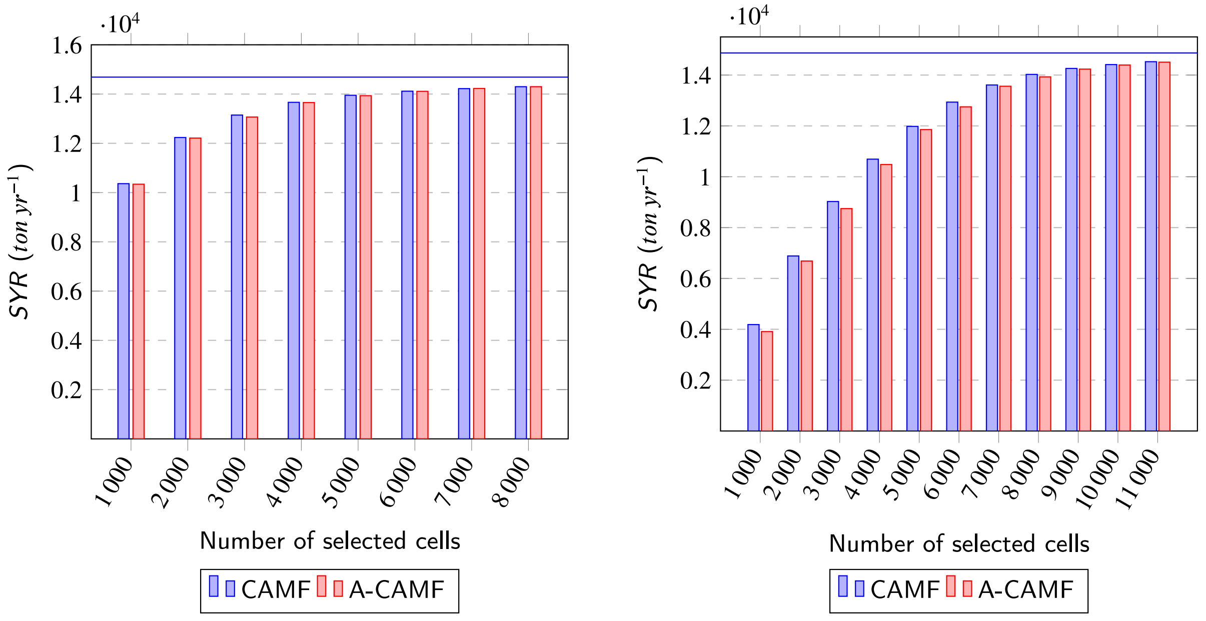


Fig. 14. Relative spatial coincidence (RSC) after afforesting 5% and 10% of the candidate cells of the Maarkeebeek data-set, when A-CAMF is used with $K = 10$ and $N = 2\,000$ for different values of T. Acceleration 1: when multiple cells are selected per iteration; Acceleration 1+2: when additionally, partial ranking is used.

comparison of the sediment production maps α^1 for Tabacay (Fig. 2(b)) and Maarkebeek (Fig. 4(b)) shows that in Tabacay, these cells are concentrated in the lowest region of the catchment, with a high probability of sharing the same path to the outlet. In Maarkebeek, these cells are distributed throughout the whole catchment, following various paths to the outlet. Using CAMF, (tentatively) afforesting cell i in iteration k has the same effect on sediment yield reduction (SYR) as in iteration $k-1$, in which cell j was selected, except when cells i and j share partly coinciding flow paths, see Section 2. Consequently, if many high-ranked cells are interconnected via flow paths, as in Maarkebeek, their ranking can substantially change in subsequent CAMF iterations. This change in ranking is not taken into account when A-CAMF selects multiple cells in one iteration, resulting in significant differences in selected cells compared to CAMF and causing a large RD. In Tabacay, fewer high-ranked cells are interconnected via flow paths, resulting in fewer changes in the ranking in CAMF and thus a smaller RD when A-CAMF selects multiple cells. We conclude that differing hydrological configurations and/or differing spatial distribution of candidate cells with



(a)   (b)

Fig. 15. Convergence of CAMF and A-CAMF heuristic methods and evolution of SYR (ton yr^{-1}). Horizontal lines: maximum attainable SYR by afforesting all candidate cells in: (a) Tabacay and (b) Maarkebeek.

high potential for *SYR* lead to different effects of spatial interaction in CAMF and A-CAMF, resulting in larger or smaller differences in solution quality.

However, when suitable values for the parameters T, K and N are chosen, *RD* is less than a few percent — see Figs. 10 and 14, and Table 7. Consequently, the error resulting from applying the accelerated version of CAMF is probably smaller than the typical modeling error, due to uncertainties in the erosion model as described in Larsen and MacDonald (2007) and Benavidez et al. (2018).

5.2. Analysis of algorithmic speedup

From Figs. 8 and 12, representing the total speedup on 28 cores (TS^T_{28}) and Fig. 6, indicating that the speedup due to parallelization varies between 21 and 24, we observe that the speedup due to the algorithmic adaptations varies drastically with the selected parameters, ranging from 19 to 1 959 for the Tabacay catchment and from 8 to 768 for the Maarkebeek catchment (see tables in Appendix D).

As discussed in Section 4.1, in the first iteration(s), cells with the largest potential for sediment yield reduction (*SYR*) are selected, while in later iterations, the differences in potential for SYR for highly-ranked cells are small. Hence, in later iterations, more cells are selected since T is fixed during the iterations and the algorithmic speedup obtained with A-CAMF increases when more cells are afforested. However, the number of cells selected per iteration, and thus the algorithmic speedup, also depend on the properties of the catchment. For example, comparing the number of iterations needed to afforest 5% and 10% of the candidate cells in Tables D.1 and D.2 in Appendix D, shows that, for $T = 0.1$, only 8 iterations are needed to select at least 1362 cells in the Tabacay catchment, while for Maarkebeek 17 iterations are needed to select 2689 cells.

5.3. Optimality, convergence and robustness of the optimization heuristics

To assess the optimality and the convergence of the optimization heuristics in CAMF and A-CAMF, we simulated the afforestation of all candidate cells in both study regions, and monitored the evolution of *SYR* in function of the number of selected cells, Fig. 15. For CAMF we observed fast convergence to SYR_{max} (all candidate cells afforested): SYR_{max} is nearly reached after selecting ≈4000 cells (14.7% of the candidate cells) in the Tabacay catchment, and ≈7000 cells (9.5% of the candidate cells) in the Maarkebeek catchment. The convergence of A-CAMF is similar to the convergence of CAMF, but fewer iterations are required to achieve approximately SYR_{max}: 50 and 41 iterations to select 4000 and 7000 cells in Tabacay and Maarkebeek respectively.

The optimization heuristic in CAMF is robust since no parameters must be set. In A-CAMF, three parameters must be set, but we presented a simple yet robust tuning procedure. Other optimization heuristics typically involve multiple parameters, such as the specific GA used in Domingues et al. (2020) for a similar problem. To find the optimal sites for afforesting $9\,331 = 20\%$ of the raster cells, the GA begins with an initial population of rasters with 20% afforested cells, randomly allocated. The GA operators alter the positions of the afforested cells to optimize the fitness function, based on the USPED erosion/deposition model. Several scenarios for the GA are used by varying the values of the GA parameters: population size and number of iterations. The solution quality strongly depends on the scenario. The best result is obtained with population size 500 and 60 000 iterations. In this scenario, the USPED model is executed $3 \cdot 10^7$ times, with no guarantee to be close to the optimal solution.

Several studies (Domingues et al., 2020; Cibin and Chaubey, 2015) propose evolutionary algorithms for solving spatial optimization problems similar to the one addressed in this paper. Our results indicate that the adapted steepest ascent hill climbing method is a valid and efficient alternative solution suitable for this type of problem. For example, CAMF and A-CAMF have been used to identify locations for land cover changes to minimize downstream river flood hazard (Gabriels et al., 2021, 2022), minimize loss of radioactively contaminated sediment in a watershed through afforestation (Abrams et al., 2023) and maximize reduction of runoff discharge volume and peak discharge at the outlet of a watershed (Rosier et al., 2024c).

5.4. Limitations

The results obtained with CAMF and A-CAMF strongly depend on the soil erosion and sediment transport models used. In this paper, we used the RUSLE model (Renard et al., 1991) for soil erosion. To obtain accurate predictions of erosion rates for different land covers, the model parameters should be tested and calibrated for the specific area (Issazadeh et al., 2012). Similarly, the use of a piece-wise linear function based on retention capacity and saturation threshold to describe sediment transport between cells is a simplification of a more complex process involving other environmental factors (Wainwright et al., 2015).

However, other models to estimate soil erosion can be used in CAMF/A-CAMF, as the sediment production map is computed offline beforehand, and then entered as an input for these methods. Additionally, the calculation of the sediment accumulation (*SA* matrix) is organized in a manner that allows to easily plug in other models to simulate sediment transport, such as the model based on transport capacity proposed by Van Oost et al. (2000) and Van Rompaey et al. (2001). Nevertheless, as mentioned in e.g. Domingues et al. (2020), to estimate erosion risk levels and plan mitigation measures, understanding patterns within the landscape is more important than obtaining the exact values.

Finally, the CAMF and A-CAMF optimization heuristics are limited to a single objective, neglecting other objectives that can also be important in spatial optimization problems, such as financial cost or spatial relations in the selection process, as in Domingues et al. (2020). While the methods have already been adapted to select contiguous or compact sites in Vanegas et al. (2014), Castillo-Reyes et al. (2023b) and Castillo-Reyes et al. (2023), how multi-objective optimization strategies can be incorporated into the A-CAMF software requires further exploration.

6. Conclusions

The effectiveness of interventions aimed at minimizing damage at target sites within a hydrological catchment is highly location-specific, due to the spatial variability of soil erosion intensity and the spatial interaction in sediment transportation processes. The CAMF method selects in a raster representation of the catchment those cells where an intervention, such as afforestation, minimizes sediment loss at the outlet, under given constraints. This spatial optimization problem is solved by a steepest ascent hill climbing method, providing a robust and fast converging alternative to exact and other heuristic methods. However, the computational cost of CAMF substantially increases with increasing problem size and hence may become prohibitive for large catchments.

In this paper, we introduced A-CAMF, which incorporates adaptations to the CAMF selection algorithm and efficient parallelization for multi-core processors, substantially reducing execution time without compromising the solution quality. Results obtained for two river catchments with different properties illustrate that A-CAMF can be several orders of magnitude faster than CAMF, depending on the strength of the spatial interaction in the catchment and the values of the parameters that control the adaptations. While sequential CAMF software may require hours or days to solve the test cases discussed herein, A-CAMF can solve these problems in less than 1 h on a multi-core computer with 28 cores, producing outcomes that are largely similar to those of the original CAMF, in terms of both cells to be afforested and magnitude of the sediment loss reduction. Furthermore, the results indicate that higher levels of spatial interaction within the catchment result in a greater impact on the relative accuracy of A-CAMF versus CAMF. We also developed a tuning algorithm to automatically select suitable values for the parameters that control the adaptations of the optimization algorithm in CAMF. Executing the tuning procedure and subsequently A-CAMF with the tuned parameters is still substantially faster than executing CAMF.

In conclusion, the CAMF/A-CAMF optimization approach is suitable for real-world intervention planning applications encompassing not only the minimization of sediment loss, water discharge, or export of contaminants from a catchment but also the minimization of accumulation of sediment, water, or contaminants at target sites within the catchment. In future research, A-CAMF will be extended to locate sites surrounding urban areas in mountainous catchments where deforestation should be avoided to minimize negative effects at city borders.

Code availability

The source code is available in a public GitLab repository: https://gitlab.kuleuven.be/u0123674/acamf

CRediT authorship contribution statement

Grethell Castillo-Reyes: Writing – review & editing, Writing – original draft, Software, Investigation, Formal analysis, Conceptualization. **René Estrella:** Validation, Resources, Formal analysis, Data curation. **Dirk Roose:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Formal analysis, Conceptualization. **Floris Abrams:** Resources, Data curation. **Gerdys Jiménez-Moya:** Supervision. **Jos Van Orshoven:** Validation, Supervision, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The source code is available in a public GitLab repository: https://gitlab.kuleuven.be/u0123674/acamf.

Acknowledgments

The authors acknowledge the support of the VLIR-UOS projects "Afforestation and avoidance of deforestation to protect mountain cities from floods and mud flows" and "Networks 2019 Phase 2 Cuba ICT" (PhD research grant of G. Castillo-Reyes), and the "Global Minds KU Leuven" programme (Research Stay of R. Estrella).

Appendix A. Sediment accumulation

The amount of sediment D_i leaving cell i is determined as in Vanegas (2010) and Castillo-Reyes et al. (2023a). If SA_i is below the retention capacity, no sediment flows into neighboring cells. If SA_i is between the retention capacity and the saturation threshold, a fraction, denoted as the flow factor, of the sediment not retained in the cell flows into down-slope cells. If SA_i is above the saturation threshold, the amount of sediment above the threshold is fully delivered to down-slope cells, i.e.

$$D_i = \begin{cases} 0, & \text{if } SA_i \leq \rho_i^k \\ \gamma_i^k (SA_i - \rho_i^k), & \text{if } \rho_i^k < SA_i \leq \sigma_i^k \\ \gamma_i^k (\sigma_i^k - \rho_i^k) + (SA_i - \sigma_i^k), & \text{if } SA_i > \sigma_i^k \end{cases} \tag{A.1}$$

The amount of sediment flowing to a down-slope neighbor cell j is given by

$$D_{i,j} = D_i \times F_{i,j} \tag{A.2}$$

with $F_{i,j}$ the fraction of sediment flowing from i to j, calculated by the flow direction model used.

Appendix B. Algorithm for the hyper-parameter tuning process

Algorithm B.1 Selection of appropriate parameters for A-CAMF

```
Input: Number of cells to be used for tuning $m$, maximum relative difference $RD_{max}$ allowed by the user

1. $t \leftarrow 0$
2. $k \leftarrow 0$
3. $SYR_{CAMF} \leftarrow$ Compute original CAMF to select $m$ cells.

repeat
    4. $T = S_t^T$
    5. $SYR_{A\text{-}CAMF} \leftarrow$ Compute A-CAMF to select $m$ cells with $T$, $K = 0$ and $N =$ number of candidate cells.
    6. $RD_m \leftarrow$ Compute relative difference between $SYR_{CAMF}$ and $SYR_{A\text{-}CAMF}$

    if $RD_m > RD_{max}$ and $S_{t+1}^T \neq 0$ then
        7. $t = t + 1$
    end if

    if $RD_m > RD_{max}$ and $S_{t+1}^T = 0$ then
        8. A-CAMF cannot select multiple cells per iteration. Continue with acceleration 2 with $T = 0$.
    end if
until $RD_m > RD_{max}$

repeat
    9. $K = S_k^K$
    10. Select $N$ in function of $K$ and the $R$ cells selected in the last iteration with A-CAMF using $T$.
    11. $SYR_{A\text{-}CAMF} \leftarrow$ Compute A-CAMF to select $m$ cells with $T$, $K$ and $N$.
    12. $RD_m \leftarrow$ Compute relative difference between $SYR_{CAMF}$ and $SYR_{A\text{-}CAMF}$

    if $RD_m > RD_{max}$ and $S_{k+1}^K \neq 0$ then
        13. $k = k + 1$
    end if

    if $RD_m > RD_{max}$ and $S_t^T = 0$ and $S_{k+1}^K = 0$ then
        14. Accelerations cannot be activated.
    end if
until $RD_m > RD_{max}$

Output: Set of parameters for A-CAMF ($T$, $K$, $N$)
```

Appendix C. Parameter values used in the sediment transport model

See Tables C.1 and C.2.

Table C.1
Parameter values used in the sediment transport model for the Tabacay data-set (Estrella, 2015; Castillo-Reyes et al., 2023a).

Parameter	Initial value (before afforestation)	Second value (after afforestation)
Sediment production	α^1, calculated by RUSLE, Fig. 2(b)	$\alpha^2 = 0.83 \times \alpha^1$
Retention capacity	$\rho^1 = 0.37 \times \alpha^1$	$\rho^2 = 0.61 \times \alpha^1$
Saturation threshold	$\sigma^1 = 0.96 \times \alpha^1$	$\sigma^2 = 0.98 \times \alpha^1$
Flow factor	γ^1, normalized slope from DEM	$\gamma^2 = 0.75 \times \gamma^1$

Table C.2
Parameter values used in the sediment transport model for the Maarkebeek data-set (Castillo-Reyes et al., 2023a).

Parameter	Before afforestation	After afforestation
Sediment production	α^1, calculated by RUSLE, Fig. 4(b)	$\alpha^2 = 0.83 \times \alpha^1$
Retention capacity	$\rho^1 = 0.55 \times \alpha^1$	$\rho^2 = 0.73 \times \alpha^1$
Saturation threshold	$\sigma^1 = \alpha^1$	$\sigma^2 = 1.02 \times \alpha^1$
Flow factor	γ^1, normalized slope from DEM	$\gamma^2 = 0.75 \times \gamma^1$

Appendix D. Performance of A-CAMF for different data-sets

The algorithmic speedup AS_p^T and the total speedup TS_p^T presented in the following tables are defined as

$$AS_p^T = \frac{T_p}{T_p^T}; \quad TS_p^T = \frac{T_s}{T_p^T} \tag{D.1}$$

where T_s and T_p are the sequential and parallel execution times of CAMF and T_p^T is the parallel time on p cores of A-CAMF with threshold $T \geq 0$.

Appendix E. Definition of variables and acronyms

List of Abbreviations

A-CAMF	Accelerated-CAMF
CAMF	Cellular Automata based heuristic for Minimizing Flow
DEM	Digital Elevation Model
GA	Genetic algorithm
GAs	Genetic algorithms
IP	Integer Programming
LP	Linear Programming
MFD	Multiple Flow Direction
MFD-D∞	Multiple Flow Direction-DInfinity
MFD-FD8	Multiple Flow Direction-Fractional Deterministic Eight Neighbor
RD	Relative Difference
RSC	Relative Spatial Coincidence
RUSLE	Revised Universal Soil Loss Equation
SA	Sediment Accumulated
SFD	Single Flow Direction
SY	Sediment Yield

Table D.1
Performance of A-CAMF for the Tabacay data-set when multiple cells are selected per iteration and partial ranking is used with $K = 20$, $N = 1500$. T: threshold value; RD: relative difference between the results of CAMF and A-CAMF (in %); T_{28}^{T}: CPU time (s) on 28 cores; AS_{28}^{T}: Algorithmic Speedup; TS_{28}^{T}: Total Speedup; # afforested cells: 5%, 10%, 20% and 30% of the candidate cells.

# Afforested cells	T	# iterations	SYR (ton yr^{-1})	RD (%)	T_{28}^{T}	AS_{28}^{T}	TS_{28}^{T}
5% = 1362	0.00	1362	11 243.77	0.00	2 125	19	417
	0.01	246	11 241.81	0.02	394	104	2 253
	0.05	73	11 228.46	0.14	110	372	8 041
	0.10	41	11 202.76	0.36	64	641	13 867
10% = 2724	0.00	2724	12 944.78	0.00	4 116	20	431
	0.01	336	12 944.19	0.00	526	157	3 373
	0.05	84	12 940.38	0.03	131	628	13 537
	0.10	49	12 936.06	0.07	77	1070	23 058
20% = 5448	0.00	5448	14 034.18	0.00	8 444	20	420
	0.01	482	14 033.90	0.00	751	228	4 723
	0.05	114	14 032.73	0.01	172	996	20 596
	0.10	58	14 030.50	0.03	91	1891	39 085
30% = 8172	0.00	8172	14 310.08	0.00	12 748	19	417
	0.01	576	14 309.91	0.00	893	275	5 961
	0.05	141	14 309.12	0.01	218	1117	24 375
	0.10	81	14 308.16	0.01	126	1959	42 386

Table D.2
Performance of A-CAMF for the Maarkebeek data-set when multiple cells are selected per iteration and partial ranking is used with $K = 10$ and $N = 2000$. T: threshold value; RD: the relative difference between the results of CAMF and A-CAMF (in %); T_{28}^{T}: CPU time (s) on 28 cores; AS_{28}^{T}: Algorithmic Speedup; TS_{28}^{T}: Total Speedup; # afforested cells: 5%, 10%, 20% and 30% of the candidate cells.

# Afforested cells	T	# iterations	SYR (ton yr^{-1})	RD (%)	T_{28}^{T}	AS_{28}^{T}	TS_{28}^{T}
5% = 2690	0.00	2 689	8 409.06	0.00	48 689	8	180
	0.01	127	8 386.43	0.29	4 357	90	2 012
	0.05	41	8 284.76	1.40	1 400	279	6 262
	0.10	21	8 108.00	2.81	763	511	11 490
10% = 5379	0.00	5 374	12 374.97	0.00	99 781	8	176
	0.01	192	12 353.81	0.17	7 300	106	2 401
	0.05	51	12 267.24	0.83	1 868	414	9 385
	0.10	38	12 118.88	1.85	1 173	660	14 948
20% = 10 758	0.00	10 739	14 500.27	0.00	199 530	8	176
	0.01	411	14 498.12	0.01	15 626	99	2 243
	0.05	94	14 487.82	0.09	3 553	437	9 865
	0.10	61	14 471.33	0.19	2 267	684	15 463
30% = 16 138	0.00	16 112	14 802.24	0.00	311 769	8	169
	0.01	551	14 801.68	0.00	21 358	110	2 462
	0.05	121	14 799.30	0.02	4 616	507	11 393
	0.10	81	14 794.49	0.05	3 052	768	17 231

SYR Sediment Yield Reduction

USPED Unit Stream Power Based Erosion Deposition

List of variables

The following list describes several variables used within the body of the manuscript

α_i^k Sediment produced locally in cell i, before afforestation ($k = 1$) and after afforestation ($k = 2$).
γ_i^k Flow factor in cell i, before afforestation ($k = 1$) and after afforestation ($k = 2$).
ρ_i^k Retention capacity in cell i, before afforestation ($k = 1$) and after afforestation ($k = 2$).
σ_i^k Saturation threshold in cell i, before afforestation ($k = 1$) and after afforestation ($k = 2$).
AS^T Algorithmic speedup for a threshold T.
RD_n The relative difference in SYR values when afforesting n cells.
RD_{max} A user-defined relative difference.
SA Sediment Accumulated.
SYR Sediment Yield Reduction.
SY Sediment Yield.
SY^0 Sediment Yield at the initial situation.
TS^T Total speedup for a threshold T.
K Intermediate number of iterations in which the partial ranking is computed.
K_{step} Step value to compute the set K values used in the hyper-parameter tuning process.
N Top-N of cells of the ranking produced in the last iteration taken into account in the partial ranking acceleration.
RD Relative Difference.
S^K Set of K values used in the hyper-parameter tuning process.
S^T Set of threshold T values used in the hyper-parameter tuning process.
S_p Speedup on p CPU cores.
T Relative Threshold T.
T_p Parallel time on p CPU cores.
T_s Sequential time.
T_{max} Max value for threshold T used in the hyper-parameter tuning process.
T_{step} Step value to compute the set of threshold T values used in the hyper-parameter tuning process.

References

Abrams, F., Sweeck, L., Camps, J., Castillo-Reyes, G., Feng, B., Onda, Y., Van Orshoven, J., 2023. Minimizing the loss of radioactively contaminated sediment from the niida watershed (Fukushima, Japan) through spatially targeted afforestation. In: EGU General Assembly Conference Abstracts. pp. EGU–12670. http://dx.doi.org/10.5194/egusphere-egu23-12670.

Arabi, M., Govindaraju, R.S., Hantush, M.M., 2006. Cost-effective allocation of watershed management practices using a genetic algorithm. Water Resour. Res. 42 (10), http://dx.doi.org/10.1029/2006WR004931, URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2006WR004931. arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2006WR004931.

Bachmatiuk, J., Garcia-Gonzalo, J., Borges, J., 2015. Analysis of the performance of different implementations of a heuristic method to optimize forest harvest scheduling. Silva Fennica http://dx.doi.org/10.14214/sf.1326.

Benavidez, R., Jackson, B., Maxwell, D., Norton, K., 2018. A review of the (revised) universal soil loss equation ((R)USLE): with a view to increasing its global applicability and improving soil loss estimates. Hydrol. Earth Syst. Sci. 22 (11), 6059–6086. http://dx.doi.org/10.5194/hess-22-6059-2018, URL: https://hess.copernicus.org/articles/22/6059/2018/.

Bettinger, P., Sessions, J., Chung, W., Graetz, D., Boston, K., 2003. Eight heuristic planning techniques applied to three increasingly difficult wildlife planning problems: A summary. In: Systems Analysis in Forest Resources: Proceedings of the Eighth Symposium, Held September 27–30, 2000, Snowmass Village, Colorado, U.S.A.. Springer Netherlands, Dordrecht, pp. 249–257. http://dx.doi.org/10.1007/978-94-017-0307-9_24.

Borges, J.G., Hoganson, H.M., Falcão, A.O., 2002. Heuristics in multi-objective forest management. In: Multi-Objective Forest Planning. Springer Netherlands, Dordrecht, pp. 119–151. http://dx.doi.org/10.1007/978-94-015-9906-1_6.

Castillo-Reyes, G., Estrella, R., Gabriels, K., Van Orshoven, J., Abrams, F., Roose, D., 2023a. Selecting sites for afforestation to minimize sediment loss from a river basin: Computational complexity of single and multiple flow direction methods in raster databases. Comput. Geosci. 171, 105269. http://dx.doi.org/10.1016/j.cageo.2022.105269, URL: https://www.sciencedirect.com/science/article/pii/S0098300422002187.

Castillo-Reyes, G., Roose, D., Estrella, R., Abrams, F., Jiménez-Moya, G., Van Orshoven, J., 2023. Selection of groups of contiguous cells for sediment yield minimization through afforestation: Application in the Manicaragua region. In: Cardenas, R., et al. (Eds.), Proceedings of the IV International Conference on BioGeoSciences. UCLV, submitted for publication.

Castillo-Reyes, G., Roose, D., Estrella, R., Abrams, F., Jiménez-Moya, G., Van Orshoven, J., 2023b. Extension of A-CAMF to select groups of contiguous cells for intervention: Computational cost vs. solution quality. Rev. Cubana Cienc. Inform. URL: https://rcci.uci.cu/?journal=rcci&page=article&op=view&path%5B%5D=2851.

Chapman, B., Jost, G., Pas, R.V.D., 2007. Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation). The MIT Press.

Chichakly, K.J., Bowden, W.B., Eppstein, M.J., 2013. Minimization of cost, sediment load, and sensitivity to climate change in a watershed management application. Environ. Model. Softw. 50, 158–168. http://dx.doi.org/10.1016/j.envsoft.2013.09.009, URL: https://www.sciencedirect.com/science/article/pii/S1364815213002004.

Cibin, R., Chaubey, I., 2015. A computationally efficient approach for watershed scale spatial optimization. Environ. Model. Softw. 66, 1–11. http://dx.doi.org/10.1016/j.envsoft.2014.12.014, URL: https://www.sciencedirect.com/science/article/pii/S1364815214003715.

Domingues, G.F., Marcatti, G.E., dos Santos, A.G., Lorenzon, A.S., de Almeida Telles, L.A., de Castro, N.L.M., Barros, K.O., Gonzáles, D.G.E., de Carvalho, J.R., da Silva Gandine, S.M., da Costa de Menezes, S.J.M., dos Santos, A.R., Ribeiro, C.A.A.S., 2020. Optimized allocation of forest restoration zones to minimize soil losses in watersheds. J. Environ. Manag. 271, 110923. http://dx.doi.org/10.1016/j.jenvman.2020.110923, URL: https://www.sciencedirect.com/science/article/pii/S0301479720308525.

Dong, L., Bettinger, P., Liu, Z., Qin, H., 2015. A comparison of a neighborhood search technique for forest spatial harvest scheduling problems: A case study of the simulated annealing algorithm. Forest Ecol. Manag. 356, 124–135, URL: https://api.semanticscholar.org/CorpusID:88108140.

Estrella, R., 2015. Where to Afforest? Single and Multiple Criteria Evaluation Methods for Spatio-Temporal Decision Support, with Application to Afforestation (Ph.D. thesis). KU Leuven, URL: https://lirias.kuleuven.be/retrieve/345463.

Estrella, R., Cattrysse, D., Van Orshoven, J., 2014a. Comparison of three ideal point-based multi-criteria decision methods for afforestation planning. Forests 5 (12), 3222–3240. http://dx.doi.org/10.3390/f5123222, URL: https://www.mdpi.com/1999-4907/5/12/3222.

Estrella, R., Vanegas, P., Cattrysse, D., Van Orshoven, J., 2014b. Trading off accuracy and computational efficiency of an afforestation site location method for minimizing sediment yield in a river catchment. In: Proceedings of GEOProcessing 2014: The Sixth International Conference on Advanced Geographic Information Systems, Applications, and Services. International Academy, Research, and Industry Association (IARIA), pp. 94–100, URL: https://lirias.kuleuven.be/retrieve/267799.

Fischer, D.T., Church, R.L., 2003. Clustering and compactness in reserve site selection: An extension of the Biodiversity Management Area selection model. Forest Sci. 49 (4), 555–565. http://dx.doi.org/10.1093/forestscience/49.4.555.

Gabriels, K., Willems, P., Van Orshoven, J., 2021. Performance evaluation of spatially distributed, CN-based rainfall-runoff model configurations for implementation in spatial land use optimization analyses. J. Hydrol. 602, 126872. http://dx.doi.org/10.1016/j.jhydrol.2021.126872, URL: https://www.sciencedirect.com/science/article/pii/S0022169421009227.

Gabriels, K., Willems, P., Van Orshoven, J., 2022. An iterative runoff propagation approach to identify priority locations for land cover change minimizing downstream river flood hazard. Landsc. Urban Plan. 218, 104262. http://dx.doi.org/10.1016/j.landurbplan.2021.104262, URL: https://www.sciencedirect.com/science/article/pii/S0169204621002255.

Gersmehl, P., 1970. Spatial interaction. J. Geogr. 69 (9), 522–530. http://dx.doi.org/10.1080/00221347008981861.

Hayes, M.C., Wilson, A., 1971. Spatial interaction. Soc.-Econ. Plan. Sci. 5 (1), 73–95. http://dx.doi.org/10.1016/0038-0121(71)90042-5, URL: https://www.sciencedirect.com/science/article/pii/0038012171900425.

Heinonen, T., Pukkala, T., 2004. A comparison of one and two compartment neighbourhoods in heuristic search with spatial forest management goals. Silva Fennica 38, http://dx.doi.org/10.14214/sf.419.

Issazadeh, L., Sokouti, R., Homaee, M., Pazira, E., 2012. Comparison of empirical models to estimate soil erosion and sediment yield in micro catchments. Eurasian J. Soil Sci. 1, 28–33.

Jiménez, Á.B., Lázaro, J.L., Dorronsoro, J.R., 2007. Finding optimal model parameters by discrete grid search. In: Innovations in Hybrid Intelligent Systems. Springer, pp. 120–127.

Kaim, A., Cord, A.F., Volk, M., 2018. A review of multi-criteria optimization techniques for agricultural land use allocation. Environ. Model. Softw. 105, 79–93. http://dx.doi.org/10.1016/j.envsoft.2018.03.031, URL: https://www.sciencedirect.com/science/article/pii/S1364815217309970.

Kaini, P., Artita, K., Nicklow, J.W., 2012. Optimizing structural best management practices using SWAT and genetic algorithm to improve water quality goals. Water Resour. Manag. 26 (7), 1827–1845. http://dx.doi.org/10.1007/s11269-012-9989-0.

Karterakis, S.M., Karatzas, G.P., Nikolos, I.K., Papadopoulou, M.P., 2007. Application of linear programming and differential evolutionary optimization methodologies for the solution of coastal subsurface water management problems subject to environmental criteria. J. Hydrol. 342 (3), 270–282. http://dx.doi.org/10.1016/j.jhydrol.2007.05.027, URL: https://www.sciencedirect.com/science/article/pii/S0022169407003198.

Kim, J., 1997. Iterated Grid Search Algorithm on Unimodal Criteria. Virginia Polytechnic Institute and State University.

Kumar, M., Sahu, A.P., Sahoo, N., Dash, S.S., Raul, S.K., Panigrahi, B., 2022. Global-scale application of the RUSLE model: a comprehensive review. Hydrol. Sci. J. 67 (5), 806–830. http://dx.doi.org/10.1080/02626667.2021.2020277, arXiv:https://doi.org/10.1080/02626667.2021.2020277.

Larsen, I.J., MacDonald, L.H., 2007. Predicting postfire sediment yields at the hillslope scale: Testing RUSLE and disturbed WEPP. Water Resour. Res. 43 (11), http://dx.doi.org/10.1029/2006WR005560, URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2006WR005560. arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2006WR005560.

Li, R., 2007. Integration of GIS Techniques and Heuristic Algorithms to Address Spatial Forest Planning Issues in the Southern U.S. (Ph.D. thesis). University of Georgia.

Maier, H., Kapelan, Z., Kasprzyk, J., Kollat, J., Matott, L., Cunha, M., Dandy, G., Gibbs, M., Keedwell, E., Marchi, A., Ostfeld, A., Savic, D., Solomatine, D., Vrugt, J., Zecchin, A., Minsker, B., Barbour, E., Kuczera, G., Pasha, F., Castelletti, A., Giuliani, M., Reed, P., 2014. Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. Environ. Model. Softw. 62, 271–299. http://dx.doi.org/10.1016/j.envsoft.2014.09.013, URL: https://www.sciencedirect.com/science/article/pii/S1364815214002679.

Michalewicz, Z., Fogel, D.B., 2000. How to Solve It: Modern Heuristics. Springer.

Nguyen, X.L., Chou, T.-Y., Fang, Y.-M., Lin, F.-C., Hoang, T.V., Huang, Y., 2017. Optimal site selection for land use planning: A comparison between two approaches of fuzzy analytic hierarchy process and fuzzy analytic network process. 4, pp. 120–124, URL: http://iraj.in/journal/journal_file/journal_pdf/13-373-1503725800120-124.pdf,

O'Callaghan, J.F., Mark, D.M., 1984. The extraction of drainage networks from digital elevation data. Comput. Vis. Graph. Image Process. 28 (3), 323–344. http://dx.doi.org/10.1016/S0734-189X(84)80011-0, URL: http://www.sciencedirect.com/science/article/pii/S0734189X84800110.

Orsi, F., Church, R.L., Geneletti, D., 2011. Restoring forest landscapes for biodiversity conservation and rural livelihoods: A spatial optimisation model. Environ. Model. Softw. 26 (12), 1622–1638. http://dx.doi.org/10.1016/j.envsoft.2011.07.008, URL: https://www.sciencedirect.com/science/article/pii/S1364815211001691.

Panagopoulos, Y., Makropoulos, C., Mimikou, M., 2013. Multi-objective optimization for diffuse pollution control at zero cost. Soil Use Manage. 29 (s1), 83–93. http://dx.doi.org/10.1111/sum.12012, URL: https://bsssjournals.onlinelibrary.wiley.com/doi/abs/10.1111/sum.12012. arXiv:https://bsssjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/sum.12012.

Renard, K.G., Foster, G.R., Weesies, G.A., Porter, J.P., 1991. RUSLE: Revised universal soil loss equation. J. Soil Water Conserv. 46 (1), 30–33, URL: https://www.jswconline.org/content/46/1/30.

Rosier, I., Diels, J., Somers, B., Van Orshoven, J., 2024c. Maximising runoff retention by vegetated landscape elements positioned through spatial optimisation. Landsc. Urban Plan. 243, 104968. http://dx.doi.org/10.1016/j.landurbplan.2023.104968, URL: https://www.sciencedirect.com/science/article/pii/S0169204623002876.

Sarma, B., Sarma, A.K., Mahanta, C., Singh, V.P., 2015. Optimal ecological management practices for controlling sediment yield and peak discharge from Hilly Urban Areas. J. Hydrol. Eng. 20 (10), 04015005. http://dx.doi.org/10.1061/(ASCE)HE.1943-5584.0001154, arXiv:https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29HE.1943-5584.0001154. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29HE.1943-5584.0001154.

Shan, Y., Bettinger, P., Cieszewski, C.J., Li, R.T., 2009. Trends in spatial forest planning. Math. Comput. Nat. Resour. Sci. 1, 86–112, URL: https://api.semanticscholar.org/CorpusID:17166200.

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. Model. Softw. 118, 241–251. http://dx.doi.org/10.1016/j.envsoft.2019.05.003, URL: https://www.sciencedirect.com/science/article/pii/S1364815218311204.

Van Oost, K., Govers, G., Desmet, P., 2000. Evaluating the effects of changes in landscape structure on soil erosion by water and tillage. Landsc. Ecol. 15 (6), 577–589. http://dx.doi.org/10.1023/A:1008198215674.

Van Rompaey, A.J.J., Verstraeten, G., Van Oost, K., Govers, G., Poesen, J., 2001. Modelling mean annual sediment yield using a distributed approach. Earth Surf. Process. Landf. 26 (11), 1221–1236. http://dx.doi.org/10.1002/esp.275, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/esp.275.

Vanegas, P., 2010. A Spatially Explicit Approach to the Site Location Problem in Raster Maps with Application to Afforestation (Ph.D. thesis). KU Leuven, KU Leuven, Belgium.

Vanegas, P., Cattrysse, D., Van Orshoven, J., 2009. Integer programming (IP) formulation for minimizing sediment delivery in a watershed by reforestation of optimal sites. In: 2009 17th International Conference on Geoinformatics. pp. 1–6. http://dx.doi.org/10.1109/GEOINFORMATICS.2009.5293538.

Vanegas, P., Cattrysse, D., Van Orshoven, J., 2012. Allocating reforestation areas for sediment flow minimization: an integer programming formulation and a heuristic solution method. Opt. Eng. 13, 247–269.

Vanegas, P., Cattrysse, D., Wijffels, A., Van Orshoven, J., 2014. Compactness and flow minimization requirements in reforestation initiatives: A heuristic solution method. Ann. Oper. Res. 219 (1), 433–456.

Wainwright, J., Parsons, A.J., Cooper, J.R., Gao, P., Gillies, J.A., Mao, L., Orford, J.D., Knight, P.G., 2015. The concept of transport capacity in geomorphology. Rev. Geophys. 53 (4), 1155–1202. http://dx.doi.org/10.1002/2014RG000474, URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014RG000474. arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2014RG000474.

Wang, J., 2017. Economic geography: Spatial interaction. In: International Encyclopedia of Geography: People, the Earth, Environment and Technology. John Wiley & Sons, Ltd, pp. 1–4. http://dx.doi.org/10.1002/9781118786352.wbieg0641, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118786352.wbieg0641.

Wijffels, A., Van Orshoven, J., 2009. Contribution of Afforestation to the Enhancement of Physical and Socio-Economic Land Performance in the Southern Andes of Ecuador: Assessment, Modelling and Planning Support. Vlaamse Interuniversitaire Raad (VLIR), Leuven, Belgium.

Wischmeier, W., Smith, D., Service, U.S.A.R., Station, P.U.A.E., 1965. Predicting Rainfall-erosion Losses from Cropland East of the Rocky Mountains: Guide for Selection of Practices for Soil and Water Conservation. Agriculture Handbook, Number 282-284, Agricultural Research Service, U.S. Department of Agriculture, URL: https://books.google.be/books?id=H3wwAAAAYAAJ.

Witlox, F., 2005. Expert systems in land-use planning: An overview. Expert Syst. Appl. 29 (2), 437–445. http://dx.doi.org/10.1016/j.eswa.2005.04.041, URL: https://www.sciencedirect.com/science/article/pii/S0957417405000928.

Yang, G., Best, E.P., 2015. Spatial optimization of watershed management practices for nitrogen load reduction using a modeling-optimization framework. J. Environ. Manag. 161, 252–260. http://dx.doi.org/10.1016/j.jenvman.2015.06.052, URL: https://www.sciencedirect.com/science/article/pii/S0301479715301444.