Environmental Modelling & Software 71 (2015) 1–14

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

An optimal Cellular Automata algorithm for simulating wildfire spread

Tiziano Ghisu [a,*], Bachisio Arca [b], Grazia Pellizzaro [b], Pierpaolo Duce [b]

[a] Department of Mechanical, Chemical and Materials Engineering, University of Cagliari, Italy
[b] Institute of Biometeorology, National Research Council, Italy

ARTICLE INFO

Article history:
Received 29 December 2014
Received in revised form
3 May 2015
Accepted 4 May 2015
Available online 22 May 2015

Keywords:
Cellular automata
Optimization
Raster-based techniques
Vector-based techniques
Wildland fire spread
Tabu search

ABSTRACT

Raster-based methods for simulating wildfire spread are computationally more efficient than vector-based approaches. In spite of this, their success has been limited by the distortions that affect the fire shapes. This work presents a Cellular Automata (CA) approach that is able to mitigate the problem of distorted fire shapes thanks to a redefinition of the spread velocity, where the equations generally used in vector-based approaches are modified by means of some correction factors. A numerical optimization approach is used to find the optimal values for the correction factors. The results are compared to the ones given by two Cellular Automata simulators from the literature under homogeneous conditions. According to this work, the proposed approach provides better results, in terms of accuracy, at a comparable computational cost. The proposed approach has then been compared to Farsite, a vector-based fire-spread simulator, under realistic slope and wind conditions, producing equivalent results in a reduced computational time.

Software availability

Name of software: OPA, Optimal Cellular Automata
Developers: Tiziano Ghisu, Bachisio Arca, Grazia Pellizzaro, Pierpaolo Duce
Contact information: t.ghisu@unica.it
Hardware required: General-purpose computer
Software required: C-compiler
Programming languages: C
Availability: Contact the developers
Year first available: 2015

1. Introduction

Methods for simulating wildland fires range from purely physical models (those based on the analysis of the physics and chemistry involved in the combustion of biomass fuel and its interaction with the atmosphere) to purely empirical models (those based on a statistical regression of the observed fire behavior) (Sullivan, 2009a). As a full formulation of the equations governing a wildland fire is still not computationally feasible, a number of simplifications are often used in physical models: simplified chemistry, averaging (time-averaging or low-pass filtering), turbulence modeling, etc. Even with these assumptions, physical and quasi-physical models are in most cases several orders of magnitude slower than real-time, even on relatively large supercomputers, thus limiting their use for operational purposes (in 2007, a high-intensity fire simulation with 16 million grid cells within a 1.5 × 1.5 km domain took 0.44 processing hours for every second of simulated time (Mell et al., 2007)). On the contrary, empirical and semi-empirical models, being based on a regression of experimental data, can provide faster-than-real-time predictions to fire spread problems (reliable when caution is placed on their limits of applicability (Perry, 1998)) and have become the basis of operational fire behavior models in use today (Sullivan, 2009b).

Empirical (and quasi-empirical) simulators are based on the combination of two elements: a fire-behavior model and a fire-spread method. *Fire-behavior models* estimate spread characteristics that are important for fire suppression planning (such as rate of spread and fireline intensity) as a function of a number of independent variables (wind speed, terrain slope, fuel moisture, fuel load, fuel density, etc.). One of the most renowned is the model of Rothermel (1972, 1983), which was developed based on a large number of laboratory experiments on surface fires of varying characteristics and forms the foundation of many fire-spread simulators in the United States and in Mediterranean Europe (Finney, 2004; Lopes et al., 2002; Peterson et al., 2009). Similar models have been developed in Australia (McArthur, 1966), Canada (Van

[*] Corresponding author.
E-mail address: t.ghisu@unica.it (T. Ghisu).

http://dx.doi.org/10.1016/j.envsoft.2015.05.001
1364-8152/© 2015 Elsevier Ltd. All rights reserved.

Wagner, 1998) and other parts of the world. *Fire-spread methods* define the rules for evolving the fire perimeter across a landscape, based on the local characteristic of fuel, weather and topography.

Existing fire-spread techniques can be split into two cathegories. The first cathegory goes under the name of *vector implementation* and treats the fire perimeter as a closed curve, discretized through a number of points, each one expanding based on the given spread model and the local conditions (fuel, weather and topography). The outer shape formed by all individual fires constitutes the new perimeter, which is further discretized and expanded. This approach was first introduced as the Huygens' principle for fire-spread simulation by (Anderson et al., 1982), using an ellipse to define the local shape of new fires. The second cathegory – *raster implementation* – treats the problem by means of a group of contigous cells that can be either inactive (burnt or not burning) or active. A set of rules defines the spread mechanism from a cell to its neighbors, usually based on time-of-arrival or heat-accumulation approaches (Peterson et al., 2009).

Under homogeneous conditions (i.e. constant fuels, weather and topography), wildland fires have been observed to produce regular shapes, such as ellipses, double ellipses and ovoids. Richards (1995) provides a mathematical formulation for the spread rate as a function of the angle from the direction of maximum spread for several of the observed shapes. Green et al. (1983) compares experimental burned areas under homogenous conditions, concluding that a simple ellipse could fit fire growth data as well as other shapes. The elliptic fire-spread template of Anderson et al. (1982) is used in Finney (2004) and many other fire-spread simulators. In real conditions, wildland fires produce shapes different from the regular shapes mentioned above and require some form of spatial and temporal discretization to account for landscape and weather non-homogeneity.

The main weakness of vector-based approaches is the need for a computationally expensive algorithm for generating the convex hull fire-spread perimeter at each time step, especially in the presence of fire crossovers and unburned islands (Glasa and Halada, 2008). Approaches based on raster implementations (such as Cellular Automata) tend to be computationally more efficient, but can suffer from significant distortion of the produced fire shape: under constant wind and homogeneus landscape conditions, the heading portion of the fire perimeter tends to be angular rather than rounded (Karafyllidis and Thanailakis, 1997) due to the constraints of the grid cell restriction to eight directions of movement (Ball and Guertin, 1992). These inaccuracies lead to modified predictions even in real landscape situations (Peterson et al., 2009). Cellular Automata methodologies have been already exploited successfully for the simulation of many macroscopic phenomena (beside from the simulation of fire spread), such as lava flows (Rongo et al., 2008), debris flows (D'Ambrosio et al., 2007), density currents (Salles et al., 2007), soil erosion (D'Ambrosio et al., 2001), crystalline etching (Zhu and Liu, 2000), topology optimization (Zakhama et al., 2009; Du et al., 2013) and traffic simulations (Zhou and Mi, 2012; Zhou et al., 2013).

A number of authors make use of larger neighborhoods, thus increasing the number of fire spread directions allowed (French et al., 1990) and mitigating the distortion of the theoretically elliptic shape. The work of Finney (2002) demonstrates how the use of an indefinitely large neighborhood leads to results equivalent to the ones given by a vector implementation, at the cost of larger computational costs and, more importantly, of large errors in the presence of real landscapes characterized by fuel and weather variability. Alternative discretizations have also been used: Frandsen and Andrews (1979) and Hernández Encinas et al. (2007) employed hexagonal cells suffering from the same distortion problems found in the square model, Johnston et al. (2008) employed an irregular grid to avoid directional biasing in fire-front propagation. To improve the produced fire perimeter, they modified the rate of spread with the inclusion of two terms: the first one divides a region where the backing rate of spread is employed from one where a variable rate of spread is used, the latter increases the maximum rate of spread to allow for the fact that on an irregular grid fire cannot travel in a straight line, but follows an irregular path dictated by the cells' connections. The first factor is chosen from experience, while for the second factor the authors use a parametric search aimed at matching the rate of spread in the direction of maximum propagation. The results for the maximum rate of spread are promising, but little insight is given on the mitigation of the elliptic shape distortion. Trunfio et al. (2011) suggest an improved algorithm for simulating wildfire spread through Cellular Automata, which, not restricting the ignition points' locations to the cell centers, does not limit the fire spread to the eight directions (or more if larger neighborhoods are used) defined by the raster discretization, producing shapes more closely resembling the theoretically elliptic shape. This is obtained by means of a local ellipse that is expanded for each cell defined by the Cellular Automata to the neighboring cells, making effectly the method a hybrid between the raster- and vector-based formulations. The results, especially when using a 24-cell neighborhood, are promising, and the additional computational cost is affordable.

Recently, GDi Gregorio et al. (2013) and Sousa et al. (2012) presented the porting of raster type fire propagation models to graphical processing unit (GPU) architectures, reporting speed-ups of over 200× against serial CPU execution. The same would not have been possible with a vector-based approach, as this method implies several branching options during the algorithms execution, making it less suitable for parallel (and massively parallel in the case of GPUs) architectures. The application is interesting and can solve fire-spread problems in the presence of uncertain parameters (which require the solution of several hundreds or thousands of deterministic simulations) faster than real-time, fundamental for application in fire fighting methodologies. However, the simulated fire shapes present the distortions typical of raster-methods, which reduce the accuracy of the results.

In recent years, a number of authors presented the development of integrated systems for faster-than-real-time simulation of wildland fires, integrating high-fidelity information of fire behavior inputs (fuel characteristics, orography and wind), fire-behavior and fire-spread simulators, and a user-friendly visualization system. The success of these tools depends both on the availability of reliable fire propagation inputs (wind, fuel, orography) and of a fast and reliable fire-spread simulator. FLogA (Bogdos and Manolakos, 2013) is an interactive web-based software for simulating and geo-animating the progression of fires occurring anywhere in Europe. All necessary data layers are generated automatically around fire ignition locations (wind is considered constant in space and equal to the value reported by the closest METAR station), while the visualization is done by means of KML layers in a Google-Earth environment. The authors make use of the publicly available fireLib library (Belvins, Apr 2015), based on a CA algorithm. Their results suffer from the common “resistance in propagating fire to any other than the 8 main propagation paths” and the fire shapes present the typical angular front of CA solvers. Cencerrado et al. (2014) present a similar system, that in addition integrates the capability of adapting uncertain fire-behavior inputs (e.g. fuel characteristics, wind) to provide a more reliable prediction of fire spread. This is achieved by means of an evolutionary optimization of fire-behavior parameters and therefore requires the simulation of a large number of possible scenarios, while the actual fire-spread is calculated with the vector-based solver Farsite (Finney, 2004). The approach is interesting and could benefit from the use of more efficient fire-spread simulation software, such as the ones based on a CA approach, but this needs to be achieved without decreasing the accuracy provided by vector-based simulators.

This work presents a Cellular Automata approach to fire propagation that allows the simulated fire perimeters to approximate the expected elliptic fire shapes more closely than other similar techniques. This is obtained without the need for complex hybrid algorithms such as in (Trunfio et al., 2011) and for larger neighborhoods such as in (Finney, 2002; Peterson et al., 2009; Trunfio et al., 2011), but through a simple modification of the equations defining the rate of spread by means of five corrections factors, whose value is defined off-line solving an optimization problem. The results are compared to the ones obtained by other cell-based simulators based on a Cellular Automata paradigm (Peterson et al., 2009; Trunfio et al., 2011) and show a significantly improved accuracy, in the presence of a comparable computational cost. The proposed approach has then been compared to Farsite (Finney, 2004), a vector-based fire-spread simulator, under realistic slope and wind conditions, producing equivalent results in a much reduced computational time.

The rest of this paper is organized as follows. Section 2 summarizes the semi-empirical equations of Rothermel (1972) and Richards (1995), used as the basis for this work, and the basic CA algorithm, taken from (Peterson et al., 2009). Section 3 presents the modifications made to the basic CA and the optimization approach used to set the optimal values for the correction factors, together with a Radial Basis Function model used to calculate the values for the correction factors during run-time. Section 4 presents the results of the optimization, analyzes the predicted fire perimeters, both for homogeneous and realistic spread conditions. Section 5 draws some conclusions and outlines future directions.

2. Fire-spread model

The fire-spread model is based on Rothermel's surface fire spread model (Rothermel, 1972). This choice was dictated by its application to many fuels around the world, such as logging slash, grasslands and shrublands and by its use in different software for fire behavior prediction such as FIREMAP (Ball and Guertin, 1992), Farsite (Finney, 2004), FireStation (Lopes et al., 2002), FlamMap (Finney, 2006), and HFIRE (Peterson et al., 2009).

A list of input parameters required by the fire propagation model is given in Table 1. The model's outputs are given in Table 2. Fuel properties are generally supplied separately for live and dead fuels, for different size classes. Properties are averaged between size classes using the surface-to-volume ratio as weighting factor, while live and dead fuels are treated separately. A full derivation and description of the parameters in use is given in (Rothermel, 1972).

I_R and R_0 represent reaction intensity and rate of spread on a flat terrain with no wind. In the presence of wind and slope, these two parameters are multiplied by $(1+\phi_w+\phi_s)$, where ϕ_w and ϕ_s are two fire-intensification factors which depend on local wind and terrain slope. The direction of maximum spread is obtained through a vectorial summation of the dimensionless coefficients for wind speed and slope (Finney, 2004). The two-dimensional fire-spread rule used is the one described in (Alexander, 1985), subsequently adapted by Finney (2004). The ellipse length-to-width ratio can be calculated as:

$$LW = 0.936e^{50.5U_{eq}} + 0.461e^{-30.5U_{eq}} - 0.397 \tag{1}$$

where the equivalent wind speed U_{eq} (in ms^{-1}) is the wind speed that alone would produce the combined effect of actual wind and terrain slope. Assuming the fire origin to be located at the rear focus of the ellipse, the fire-spread rate can be calculated as a function of the angle θ, measured from the direction of maximum spread.

$$R(\theta) = R_0\frac{1-\overline{E}}{1-\overline{E}\cos(\theta)} \tag{2}$$

where $\overline{E}$ is the ellipse eccentricity, defined as $\sqrt{1-1/LW^2}$.

2.1. Fire expansion by means of a Cellular Automata

The fire-spread algorithm has been developed based on the work of Peterson et al. (2009).

Given a method for computing the rate of fire spread (Section 2) in any direction and for determining an appropriate time step (Section 2.2) from the fastest spreading component of the fire, a state machine is used to track the movement of the fire through the cells in the simulation domain. At any instant in the simulation, all cells in the simulation domain are assigned one of four possible states:

- Cell is unburnable [U].
- Cell is flammable, but not currently ignited [N].
- Cell is flammable and is ignited, but fuel is not yet consumed [I].
- All fuel in cell has been consumed by the fire [C].

At the start of the simulation, all cells are in the unburnable [U] or not-currently-ignited [N] states. During the simulation, a cell can transition from the [N] state to the [I] state either through an independent ignition (new fire) or through the spread of a fire from an adjacent cell. This is achieved as follows: two arrays are associated to every cell that is in the [I] state, one used to accumulate the distance that the fire has traveled in each of the possible directions (8 in the presence of a Cartesian grid), the second used to store the distance between adjacent cells in each direction. When the accumulated distance in a direction exceeds the terrain distance in that direction, then the adjacent cell in that direction is transitioned from the [N] state to the [I] state.

Cells located in the interior portion of an expanding fire (i.e. which have all adjacent cells in the [I, U] or [C] state) are transitioned to the [C] state.

2.2. Time-step definition

The time step of the simulation is determined to ensure that the fastest expanding fire does not travel more than the distance between two adjacent cells within each time step:

$$\max\left(\frac{R\Delta t}{\Delta x}\right) \leq 1 \tag{3}$$

Table 1
Input parameters required by Rothermel's model (Rothermel, 1972).

Description	Symbol	Units
Fuel load	W_0	kg m^{-2}
Fuel depth	δ	m
Surface-to-volume ratio	σ	m^{-1}
Fuel heat content	h	kJ kg^{-1}
Fuel moisture content	M_f	–
Fuel moisture content of extinction	M_x	–
Ovendry particle density	ρ_p	kg m^{-1}
Fuel particle total mineral content	S_T	–
Fuel particle effective mineral content	S_e	–
Midflame-height wind speed	U_m	m s^{-1}
Terrain slope	ϕ	degrees

Table 2
Outputs from Rothermel's model (Rothermel, 1972).

Description	Symbol	Units	Equation
	A	–	$A = \frac{1}{4.239\sigma^{0.1} - 7.27}$
	B	–	$B = 0.0133\sigma^{0.54}$
	C	–	$C = 7.47exp(-0.0693\sigma^{0.55})$
	E	–	$E = 0.715exp(-1.079 \times 10^{-4}\sigma)$
Maximum reaction velocity	Γ'_{max}	s^{-1}	$\Gamma'_{max} = \frac{\sigma^{1.5}}{(1171.27 + 3.564\sigma^{1.5})}$
Optimum reaction velocity	Γ'	s^{-1}	$\Gamma' = \Gamma'_{max}\left(\frac{\beta}{\beta_{op}}\right)^{A} exp\left(\left[A\left(1 - \frac{\beta}{\beta_{op}}\right)\right]\right)$
Optimum packing ratio	β_{op}	–	$\beta_{op} = 3.348\sigma^{-0.8189}$
Moisture damping coefficient	η_M	–	$\eta_M = 1 - 2.59\frac{M_f}{M_x} + 5.11\left(\frac{M_f}{M_x}\right)^2 - 3.52\left(\frac{M_f}{M_x}\right)^3$
Mineral damping coefficient	η_S	–	$\eta_M = 0.174S_e^{-0.19}$
Propagating flux ratio	ξ	–	$\xi = \frac{exp[(0.792 + 0.376\sigma^{0.5})(\beta + 0.1)}{192 + 0.0791\sigma}$
Net fuel loading	W_n	Kg m^{-2}	$W_n = \frac{W_0}{1 + S_T}$
Ovendry bulk density ρ_b	kg m^{-3}		$\rho_b = \frac{W_0}{\delta}$
Effective heating number	ε	–	$\varepsilon = exp(-4527.56\sigma^{-1})$
Heat of preignition	Q_{ig}	kJ kg^{-1}	$Q_{ig} = 522 + 2332M_f$
Packing ratio	β	–	$\beta = \frac{\rho_b}{\rho_p}$
Reaction intensity	I_R	kJ m^2 s^{-1}	$I_R = \Gamma' W_n h \eta_M \eta_S$
Rate of spread	R_0	m s^{-1}	$R_0 = \frac{I_R\xi}{\rho_b \varepsilon Q_{ig}}$
Wind factor	ϕ_w	–	$\phi_w = CU^B\left(\frac{\beta}{\beta_o p}\right)^{-E}$
Slope factor	ϕ_s	–	$\phi_s = 5.275\beta^{-0.3}(\tan\phi)^2$

2.3. Sloping terrain

The equations introduced in Section 2 provide the rate of spread relative to the ground surface. Therefore, in the presence of sloping terrain, the value needs to be corrected to represent the rate of spread in the horizontal coordinate system.

$$R_x^h = R_x \cos(\delta_x) \quad (4)$$

$$R_y^h = R_y \cos(\delta_y) \quad (5)$$

where R_x and R_y are the rate of spread components in a coordinate system parallel to the ground, R_x^h and R_y^h are the rate of spread components in a horizontal coordinate system and δ_x and δ_y the slope components.

3. Model

Fig. 1 presents a comparison of the expected fire shapes (calculated with the elliptic model of (Alexander, 1985)) with the ones calculated with the CA formulation described in Section 2.1, under homogeneous conditions (no slope and constant wind of the intensity required to obtain the desired ellipse eccentricity). The burning time has been fixed in each simulation so to have the same distance (1800 m) covered by the fire-head. A Cartesian grid with a constant grid spacing of 20 m has been used. Equation (2) was used to calculate the spread-rate as a function of the angle between the direction of maximum spread and the spread-direction, as in other raster-based (Peterson et al., 2009; Trunfio et al., 2011) solvers. As the ellipse eccentricity increases, the predicted shape becomes more distorted, with both an increase in the length-to-width ratio (relative to the expected shape) and the formation of an angular front in the head-of-fire region (not postulated by Alexander's formulation). This behavior is due to the constraints imposed from the numerical grid on the spread directions. Peterson et al. (2009) tried to mitigate the first problem (reduced length-to-width ratio) by applying a correction factor to the length-to-width ratio calculated by Alexander (1985). He used a parametric approach to study the influence of this parameter on the calculated ellipse eccentricity. Johnston et al. (2008) use two correction factors within an irregular grid raster formulation: the first one serves as a correction for the reduced head-fire rate of spread due to the grid nonlinearity, the second one is used to correct the rate-of-spread variation in the head-fire region. Trunfio et al. (2011) mitigate this problem by developing a hybrid approach where local ellipses expand from each burning cell to the neighboring cells. The distortion of fire shapes is significantly reduced, especially in the

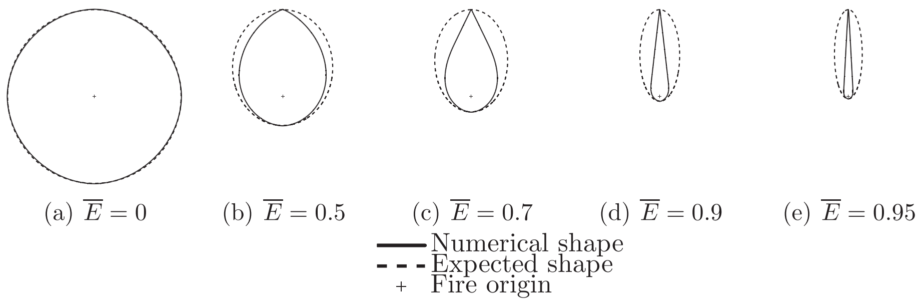

(a) $\overline{E} = 0$ (b) $\overline{E} = 0.5$ (c) $\overline{E} = 0.7$ (d) $\overline{E} = 0.9$ (e) $\overline{E} = 0.95$

Fig. 1. Comparison of expected and numerical two-dimensional fire-spread shapes under homogeneous conditions, for different eccentricities.

presence of 24-cell neighborhoods, but at the price of an increased complexity.

3.1. Modification of the fire-spread equations

This work proposes a modification of the relationship between advection velocity and spread angle – equation (2) – that aims to reduce the distortion observed in cell-based fire-spread propagation approaches. The proposed correction approach is a combination of the ones proposed by Johnston et al. (2008) and by Peterson et al. (2009). It makes use of five correction factors:

1. a factor to increase the maximum rate-of-spread (this is required when the direction of maximum spread is different from one of the grid's main axes)
2. a factor to modify the ellipse eccentricity
3. a factor to modify the dependency of the fire-spread rate on the angle from the effective wind direction in the head-fire region
4. a factor to modify the dependency of the fire-spread rate on the angle from the effective wind direction in the back-fire region
5. a factor to adjust the rear propagation speed (otherwise modified by correction factor number 1)

Table 3 presents a list of the physical quantities corrected by this method and of the modified relations (θ is the angle from the wind direction in radians, ranging from $-\pi$ to π).

3.2. Model evaluation

To assess the appropriateness of the proposed approach, a performance evaluation method must be selected: the availability of a quantitative measure of the departure from an optimal behavior (in this case the prediction of a vector-based simulator) offers the possibility of generating an automatic evaluation of the method, essential when the aim is to produce an automatic calibration through the solution of an optimization problem (Bennett et al., 2013).

The following metrics have been adopted:

$$\begin{aligned} \lambda_u &= \frac{|R-S|}{|R|} \\ \lambda_o &= \frac{|S-R|}{|S|} \\ \lambda_s &= \frac{|R{\cup}S|-|R{\cap}S|}{|R{\cup}S|} \end{aligned} \tag{6}$$

where R represents the set of cells defining the expected fire shape, S the set of cells defining the simulated fire shape and the operator $|.|$ gives the size of a set. Consequently, λ_u and λ_o represent the underpredicted and overpredicted fire area ratios, respectively, while λ_s is a measure of the level of disagreement between the two predictions. To compute the set of cells defining the expected fire shape, a cell was considered burnt when its center fell inside the fire ellipse.

The same metrics were adopted by Trunfio et al. (2011) to evaluate the performance of their Cellular Automata approach in comparison to the one of (Peterson et al., 2009). Methods for extracting more information from the comparison of fire-spread perimeters exist: (e.g. Duff et al., 2013) quantify differences between fire perimeters using linear vectors aligned with the direction of spread, producing a measure of propagation error as a function of spread-direction). In this work, the use of the error factors of equation (6) seems more appropriate, as they collapse the difference in fire perimeters in a single number that can be used effectively in an automatic optimization process.

3.3. Setting the correction factors' values

The optimal values for the correction coefficients proposed in Section 3.1 are likely to be dependent on the eccentrity of the ellipse and on the angle of maximum spread relative to the grid's main axis. Furthermore, a dependency might exist also on the grid's topography, e.g. based on rectangular or irregular cells. This work makes use of structured Cartesian grids (i.e. composed of rectangular cells), with uniform spacing in both x and y directions.

The problem of finding appropriate values for the above correction factors can be seen as a numerical optimization problem, where the aim is to find the combination of values which minimize (or maximize) an objective function (or several objective functions, if the problem is multi-objective), e.g. one (or a combination) of the metrics introduced in Section 3.2. The optimization problem can be solved by combining a numerical optimizer – an algorithm for finding the minima (or maxima) of the objective function(s) – with an evaluation tool – a system for calculating the objective function values – and a definition of the variables' space. The main advantages are the use of an intelligent search mechanism and the automation of the overall process, which allow the parameter space to be searched extensively, concentrating on the most promising areas.

A flow chart of the optimization process is presented in Fig. 2.

3.3.1. Single- and multi-objective optimization

Optimization consists in the determination of the combination of parameters (control variables) that minimize (or maximize) an objective function (*single-objective optimization*), while satisfying a number of constraints. The general form of a (single-objective) optimization problem can be expressed mathematically as:

$$\begin{array}{ll} \text{minimize} & f(x), x{\in}\mathbf{R^n} \\ \text{subject to} & c_i(\mathbf{x}) = 0,\ i = 1,2,\ldots,m' \\ & c_i(\mathbf{x}) \le 0, i = m'+1,2,\ldots,m \end{array} \tag{7}$$

where $f(\mathbf{x})$ is the objective function, $\mathbf{x}{\in}\Omega$ is the vector of the independent control variables, Ω the domain of definition (design space) and $c_i(\mathbf{x})$ is the set of equality or inequality constraints.

Table 3
Modifications to the advection velocity formulation.

Quantity Modified	Symbol	Modified equation
maximum rate of spread	R_0	$R'_0 = R_0 k_1$
ellipse eccentricity	$\overline{E}$	$\overline{E'} = \sqrt{1 - 1/LW^{k_2}}$
spread angle ($\lvert\theta\rvert < \pi/2$)	θ	$\tan(\theta') = \tan(\theta)/LW^{k_3}$
spread angle ($\lvert\theta\rvert > \pi/2$)	θ	$\tan(\theta') = \tan(\theta)/LW^{k_4}$
spread rate	$R(\theta)$	$R'(\theta') = R'_0 \frac{1-\overline{E'}}{1-\overline{E'}\cos(\theta')} - R'_0 \left(\frac{1-\overline{E'}}{1+\overline{E'}} - \frac{1-\overline{E}}{1+\overline{E}}\right) \Bigg\lvert \left(\frac{\theta'\rvert}{pi}\right)^{k_5}$

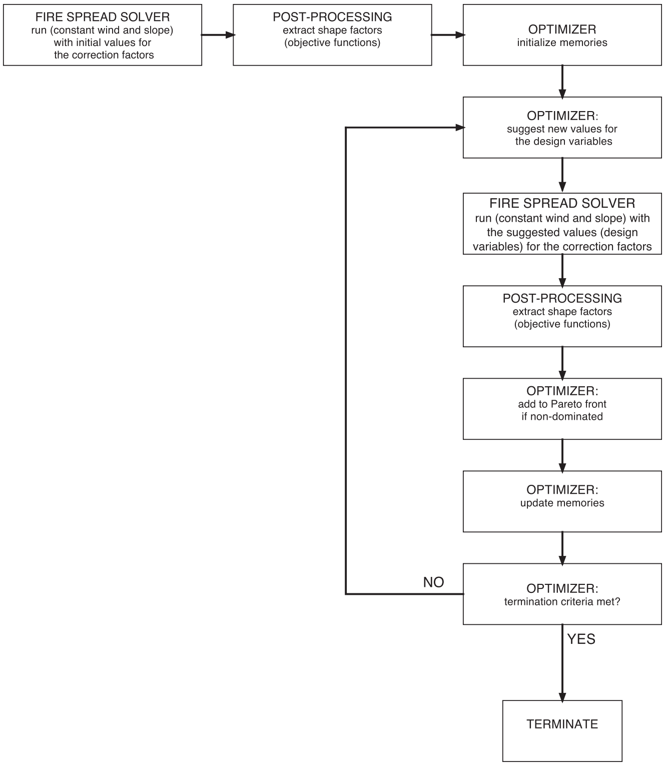


Fig. 2. Flow chart of the optimization process.

Many real-world problems involve multiple objectives that have to be optimized simultaneously (*multi-objective optimization*).

$$\begin{array}{ll} \text{minimize} & f_i(x), x \in \mathbf{R^n},\ i = 1, \dots, N \\ \text{subject to} & c_i(\mathbf{x}) = 0,\ i = 1, 2, \dots, m' \\ & c_i(\mathbf{x}) \le 0,\ i = m' + 1, 2, \dots, m \end{array} \tag{8}$$

This problem seldom admits a single perfect solution: it is most likely to have a family of alternatives that must be considered equivalent in the absence of information concerning the importance of each objective relative to the others (Fonseca and Fleming, 1995). A point $\mathbf{x}^* \in \Omega$ is *Pareto optimal* if for every $\mathbf{x} \in \Omega$ and $i = 1, \dots, N$ either

$$f_i(\mathbf{x}^*) = f_i(\mathbf{x}) \tag{9}$$

or there is at least one $i \in [1, N]$ such that

$$f_i(\mathbf{x}^*) \le f_i(\mathbf{x}) \tag{10}$$

The above definition leads to a set of solutions, usually called Pareto (optimal) front, which includes all the solutions that are non-dominated, i.e. for which no objective function can be improved without causing the degradation of at least one other objective.

3.3.2. Current optimization approach

The numerical optimizer used in this work is based on a modification of the Tabu Search (TS) approach presented by Jaeggi et al. (2008). TS is a meta-heuristic method designed to help a search negotiate difficult regions of the search space (e.g. escape from local minima) by imposing restrictions (Glover and Laguna, 1997). In Jaeggi et al. (2008)'s implementation the local search phase at its heart is conducted with the Hooke and Jeeves (H&J) algorithm (Hooke and Jeeves, 1961): a suitable increment is chosen for each variable and the value of the objective function is calculated in turn for $x_i' = x_i + \Delta_i$ and $x_i' = x_i - \Delta_i$ while keeping the other variables at their base values. The best allowed move is made. The *Short Term Memory* (STM) records the last n_{stm} visited points, which are tabu and thus cannot be revisited. This approach has been used successfully for the solution of several complex optimization problems (Ghisu et al., 2010a, 2011a, 2011b, 2011c).

Two other important features of the TS algorithm are *intensification* and *diversification*. *Intensification* is associated with the *Medium Term Memory* (MTM) where the best solutions located are stored. *Diversification* is associated with the *Long Term Memory* (LTM), which records the areas of the search space that have been searched reasonably thoroughly by dividing the design space into a number of sectors and recording how many times each sector has been visited. In the current implementation, this is achieved by dividing each design variable into $n_{regions}$ and counting the number of solutions evaluated in those regions (Jaeggi et al., 2008). On diversification the search is restarted in an under-explored region of the design space. If no improvements are made in a given iteration, a counter is incremented. After a pre-defined number of unsuccessful moves, intensification, diversification or step size reduction are performed.

In the current implementation, the MTM contains the set of non-dominated solutions found, while at every H&J move, in the absence of a single non-dominated solution, a random move is selected from among the set of non-dominated new designs. The discarded designs are not lost: they are stored in the MTM, if appropriate, and can then be selected during intensification. The concept of dominated solution has been modified, relative to Jaeggi et al.'s implementation (Jaeggi et al., 2008), adding the concept of *influence region*: a solution is superior to another only if they fall within a certain distance, otherwise the two solutions are considered equivalent (i.e. both optimal if non-dominated). The distance metric used is a weighted 1-norm distance:

$$d = \sum_{i=1}^{k} c_i \left| v_i - w_i \right| \tag{11}$$

where $\mathbf{v}$ and $\mathbf{w}$ are two design vectors of dimensionality k). This concept is particularly useful when the effect of one or more variables on the figure of merit(s) is sought.

Table 4
Optimizer's main settings.

Parameter	Description	Value
n_{stm}	Size of STM	20
$n_{regions}$	Divide search space into $n_{var} * n_{regions}$ regions	2
intensify	Perform intensification when $i_{local} = intensify$	10
diversify	Perform diversification when $i_{local} = diversify$	–
restart	Reduce step sizes and restart when $i_{local} = restart$	25
SS	Initial step-sizes (as % of variable range)	10%
SSRF	Step-sizes are multiplied by this factors at *restart*	$(0.5, 0.5, 0.5, 0.5, 0.5, 1)$

Table 4 summarizes the values chosen for the optimizer's main input parameters. The choice of appropriate values for the optimizer settings, which is clearly problem dependent, was dictated by the authors' experience (Ghisu et al., 2010b).

3.3.3. Definition of the optimization problems

The optimal values for the correction factors are likely to be dependent on the ellipse eccentricity (function of the equivalent wind) and on the direction of maximum spread. In fire-spread simulations, the ellipse eccentricity ranges between 0 and 1, while the angle between wind direction and closest grid line ranges between 0 and $\pi/4$. To limit the number of optimization runs, the optimal values for the correction factors are sought for the combination of four discrete values of ellipse eccentricity $(0.50, 0.70, 0.90, 0.95)$ and wind direction $(0, \pi/12, \pi/6, \pi/4)$. In consideration of its likely less pronounced impact on the optimal parameters' combination, the wind direction (α_w) has been treated as a variable in the optimization process, using the concept of influence region (defined in Section 3.3.2) to capture its influence on the optimal values for the k_i parameters. This allows the number of optimization runs to be significantly reduced, at the price of an increased dimensionality (larger design space) of the optimization problem. Provided that the solutions are not too dissimilar, the proposed approach should reduce significantly the computational time, while capturing the effect of wind direction on the optimal values for the correction factors. Therefore, four optimization runs were performed, for values of the effective wind speed corresponding to eccentricities of 0.50, 0.70, 0.90 and 0.95. In each simulation, the distance covered by the fire-head has been kept fixed to 1800 m, the final time has been adjusted accordingly and the time step has been fixed to half the maximum allowed by the condition in equation (3) (the time step can be adjusted adaptively for simulations in heterogeneus conditions). A computational grid with a constant grid spacing of 20 m was used in all simulations.

Two solutions have been considered to belong to different influence regions if the difference between the design variables corresponding to the wind angles (variable 6 in the design vectors) is above a trashold:

$$d = \frac{12}{\pi}\left|v_6 - w_6\right| \geq 1 \tag{12}$$

in order to obtain four solution points for different values of α_w. The vector $(1, 1, 0, 0, 1, 0)$ has been used as the initial solution point for the optimization process. The shape factor λ_s has been considered as the objective. The *single-objective optimization* problem definition is summarized in equation (13).

$$\begin{aligned} \text{Minimize} \quad & \lambda_s \\ \text{subject to} \quad & 1 \leq k_1 \leq 2 \\ & 1 \leq k_2 \leq 2 \\ & 0 \leq k_3 \leq 1 \\ & 0 \leq k_4 \leq 1 \\ & 1 \leq k_5 \leq 2 \\ & 0 \leq \alpha_w \leq \frac{\pi}{4} \end{aligned} \tag{13}$$

Despite the capability of TS to avoid local-optima within multimodal design spaces, complex landscapes increase the risk of the optimizer returning a sub-optimal solution. The use of a multi-objective approach, where the optimizer seeks to minimize (or maximize) simultaneously a number of objective functions, increases the possibilities of finding the true optimal solution(s) to a given problem. The fact that the optimizer evolves a range of solutions (rather than a single one) helps it overcome local optima: it is not uncommon for multi-objective optimization approaches to be able to find solutions that are better than the ones obtained by single-objective approaches, even relatively to the parameter used as figure of merit in the single-objective optimization.

For these reasons, the problem of finding an appropriate combination of values for the correction factors introduced in Section 3.1 has been posed mathematically as a two-objective optimization problem, with under-estimation factor λ_u and over-estimation factor λ_o as the two objectives. The *two-objective optimization* problem is summarized in equation (14).

$$\begin{aligned} \text{Minimize} \quad & \lambda_u \\ & \lambda_o \\ \text{subject to} \quad & 1 \leq k_1 \leq 2 \\ & 1 \leq k_2 \leq 2 \\ & 0 \leq k_3 \leq 1 \\ & 0 \leq k_4 \leq 1 \\ & 1 \leq k_5 \leq 2 \\ & 0 \leq \alpha_w \leq \frac{\pi}{4} \end{aligned} \tag{14}$$

3.4. A rule for calculating the correction factor in generic fire-spread conditions

To be effectively used in predictive fire-spread simulations (i.e. for generic and spatially variable values of wind speed and direction, different from the ones the corrections factors were optimized for), the proposed correction factors need to be calculated in real time (i.e. as the simulation runs) with a minimum impact on the simulation process. A model has therefore been built based on the data collected from the optimization runs. Given the relatively large number of data points, a Radial Basis Function (RBF) approach seems an appropriate choice, given their simplicity and adaptivity to highly irregular landscapes. RBFs approximate the true response as the weighted sum of m functions ϕ_j whose values depend only on the distance from a central point $\mathbf{c}_j$:

$$y^*(\mathbf{x}) = \sum_{j=1}^{m} \omega_j \phi_j\left(\left|\left|\mathbf{x} - \mathbf{c}_j\right|\right|\right) \tag{15}$$

In this work the RBF's central points have been chosen to coincide with available data points, making the model an interpolation rather than a regression. Various types of basis functions can be used; the most common ones are linear splines, cubic splines, multiquadrics, inverse multiquadrics, thin-plate splines and Gaussian functions (Golberg and Cho, 2004). This work makes use of a Gaussian function of the type

$$\phi_j\left(\left|\left|\mathbf{x} - \mathbf{c}_j\right|\right|\right) = exp\left(-\left|\left|\mathbf{x} - \mathbf{c}_j\right|\right|^2\right) \tag{16}$$

At a sample point:

$$\overline{y}_i = \sum_{j=1}^{m} \omega_j \overline{\phi}_j\left(\left|\left|\mathbf{c}_i - \mathbf{c}_j\right|\right|\right) \tag{17}$$

In matrix formulation:

$$\overline{Y} = \overline{\Phi}\Omega \tag{18}$$

where $\overline{Y}$ is the vector containing the responses at the sample points $\mathbf{c}_i$ and $\overline{\Phi}_{i,j} = \phi_j(||\mathbf{c}_i - \mathbf{c}_j||)$. At a generic point $\mathbf{x}$ the RBF prediction can be expressed as:

$$y^* = \phi^T\Omega = \phi^T\overline{\Phi}^{-1}\overline{Y} \tag{19}$$

where $\phi_j = \phi(||\mathbf{x} - \mathbf{c}_j||)$.

Following this approach, the vector of correction coefficients K^* can be calculated as:

$$K^* = \phi^T\overline{\Phi}^{-1}\overline{K} \tag{20}$$

where $\overline{K}_{i,j}$ represents the value of the correction factor k_j at the sample point i (corresponding to the i-th line in Table 8). The vector ϕ depends on $\mathbf{x}$ and needs to be calculated during run time, while the vector $\overline{\Phi}^{-1}\overline{K}$ is only a function of the initial samples and can be stored in memory.

4. Results

4.1. Single-objective optimization

Fig. 3 presents the optimization histories (objective function values as a function of the number of iterations) for the four optimization runs. Each line represents a specific wind direction. The improvements are significant and become more consistent for larger ellipse eccentricities, as the distortion becomes more pronounced (see Fig. 1). Tables 5–7 compare the results obtained with the proposed approach (OCA) – in terms of shape factor λ_s, but also of λ_u and λ_o – with the ones obtained through the application of the Cellular Automata approach without correction (BCA) and with two Cellular Automata (raster-based) simulators from the literature – the ones proposed by (Peterson et al., 2009) (named RCA) and the one proposed by (Trunfio et al., 2011) (ECA) – using both a 8-cell and a 24-cell neighborhood. The above comparison refers to the solutions obtained for $\alpha_w = 0$. There is a significant reduction in shape factor's value (−86.8% relative to the 24-cell neighborhood RCA and −37.5% relative to the 24-cell neighborhood ECA, for the 0.95 eccentricity simulation). This improvement is obtained trough a reduction in underpredicted area (the value of λ_u is reduced by 96.8% and 78.0%, respectively), while there is a marginal increase in overpredicted area, 45.0% larger than with the ECA (the value of λ_o is zero for the RCA).

Table 5
Single-objective optimization results: Shape factor λ_s under homogeneous conditions for different values of ellipse eccentricity ($\alpha_w = 0$).

$\overline{E}$	RCA-8	RCA-24	ECA-8	ECA-24	BCA	OCA
0.50	0.208	0.083	0.036	0.014	0.186	0.010
0.70	0.248	0.101	0.062	0.027	0.406	0.022
0.90	0.431	0.217	0.108	0.050	0.685	0.030
0.95	0.558	0.342	0.158	0.072	0.773	0.045

Table 6
Single-objective optimization results: Under-estimation factor λ_u under homogeneous conditions for different values of ellipse eccentricity ($\alpha_w = 0$).

$\overline{E}$	RCA-8	RCA-24	ECA-8	ECA-24	BCA	OCA
0.50	0.208	0.083	0.035	0.004	0.185	0.003
0.70	0.248	0.101	0.057	0.012	0.406	0.003
0.90	0.431	0.217	0.100	0.025	0.685	0.009
0.95	0.558	0.342	0.149	0.050	0.773	0.011

Table 7
Single-objective optimization results: Over-estimation factor λ_o under homogeneous conditions for different values of ellipse eccentricity ($\alpha_w = 0$).

$\overline{E}$	RCA-8	RCA-24	ECA-8	ECA-24	BCA	OCA
0.50	0.000	0.000	0.001	0.010	0.000	0.006
0.70	0.000	0.000	0.006	0.015	0.000	0.019
0.90	0.000	0.000	0.010	0.026	0.000	0.022
0.95	0.000	0.000	0.011	0.025	0.000	0.035

Table 8
Data used to calculate the RBF surrogate model.

$\overline{E}$	α_w	k_1	k_2	k_3	k_3	k_5
0.00	0	1.000	1.150	1.850	1.700	1.550
0.00	$\pi/12$	1.000	1.075	1.575	1.975	1.500
0.00	$\pi/6$	1.000	1.000	1.450	2.000	1.375
0.00	$\pi/4$	1.000	1.050	1.075	2.000	1.000
0.50	0	1.000	1.150	1.850	1.700	1.550
0.50	$\pi/12$	1.000	1.075	1.575	1.975	1.500
0.50	$\pi/6$	1.000	1.000	1.450	2.000	1.375
0.50	$\pi/4$	1.025	1.050	1.075	2.000	1.000
0.70	0	1.000	1.000	1.050	1.400	1.500
0.70	$\pi/12$	1.000	1.000	1.075	1.375	1.575
0.70	$\pi/6$	1.025	1.075	0.825	1.800	1.925
0.70	$\pi/4$	1.025	1.075	0.800	1.800	2.000
0.90	0	1.000	1.000	0.875	1.175	2.225
0.90	$\pi/12$	1.050	1.000	0.812	1.187	1.850
0.90	$\pi/6$	1.062	1.012	0.781	1.262	2.019
0.90	$\pi/4$	1.062	1.025	0.812	1.250	2.062
0.95	0	1.000	1.000	0.850	1.175	2.862
0.95	$\pi/12$	1.075	1.000	0.831	1.287	2.512
0.95	$\pi/6$	1.087	1.000	0.812	1.112	1.837
0.95	$\pi/4$	1.112	1.025	0.800	1.225	2.200

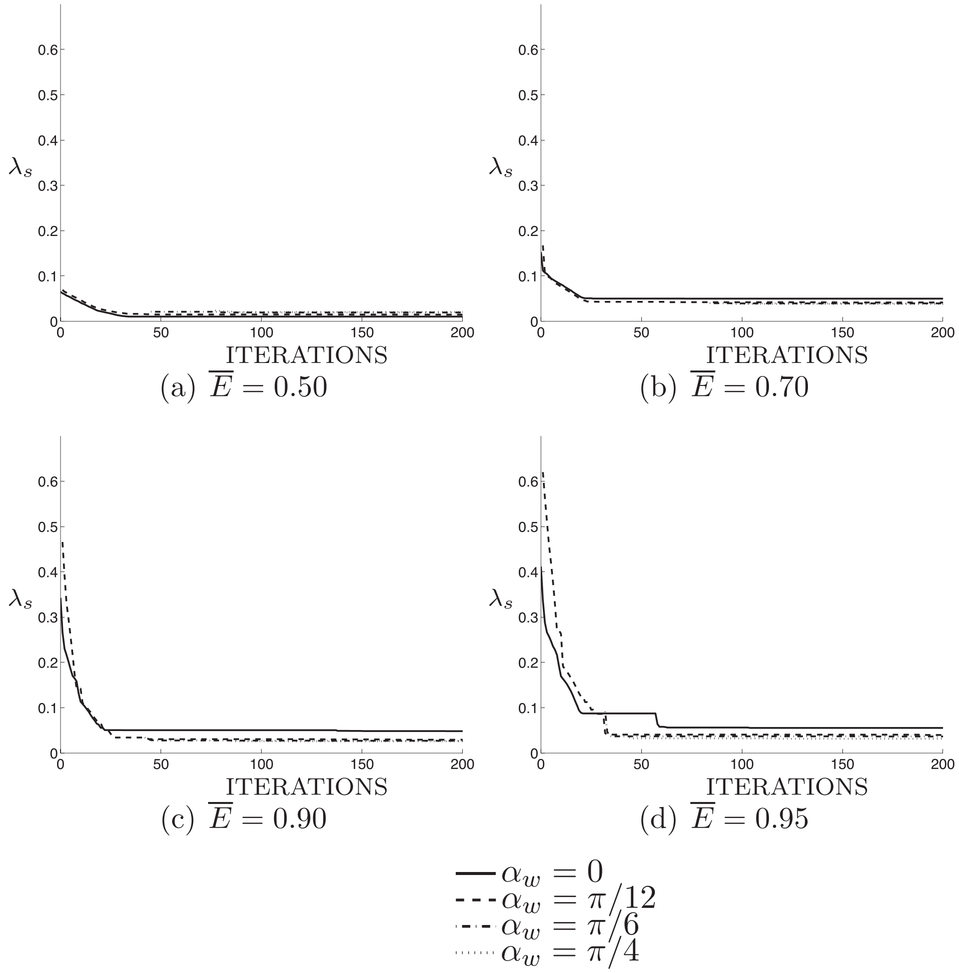
Furthermore, these results have been obtained with the use of a 8-cell neighborhood: relatively to the 8-cell neighborhood versions of the reference solvers, the improvements are even more significant. The use of a 8-cell neighborhood presents several advantages, such as the reduced complexity of the algorithm (shorter computational times), reduced memory requirements and better accuracy in the presence of rapidly changing landscapes.

4.2. Bi-objective optimization

Fig. 4 shows the Pareto fronts (i.e. the sets of non-dominated solutions) for the four optimization runs, corresponding to different values of ellipse eccentricity. It is evident how λ_u and λ_o represent two conflicting objectives (it is not possible to minimize both simultaneously, at least with the parameterization in use). The result of each optimization run is a set of trade-offs (for different values of wind direction). The most appropriate solution can be chosen from these depending on the relative importance of over- and under-predicted areas. A reasonable approach can be to select a solution lying close to the 45-degree line (i.e. with similar values for λ_u and λ_o).

4.3. Correction factors in generic fire-spread conditions

The interpolation model introduced in Section 3.4 has been used to formulate a general rule for evolving the CA with respect to time. This is required to avoid the need for solving an optimization problem for every cell (i.e. every combination of wind speed and direction) of the CA during run-time. Instead, the CA makes use of correction factors calculated with a RBF model build using the data collected from the optimization runs that were performed for discrete combinations of the two parameters. The sample points used to construct the model correspond to the results of the optimization runs, with the addition of four points corresponding to no-wind conditions, where the correction should vanish. The input data are summarized in Table 8.



(a) $\overline{E} = 0.50$

(b) $\overline{E} = 0.70$

(c) $\overline{E} = 0.90$

(d) $\overline{E} = 0.95$

Fig. 3. Optimization history.

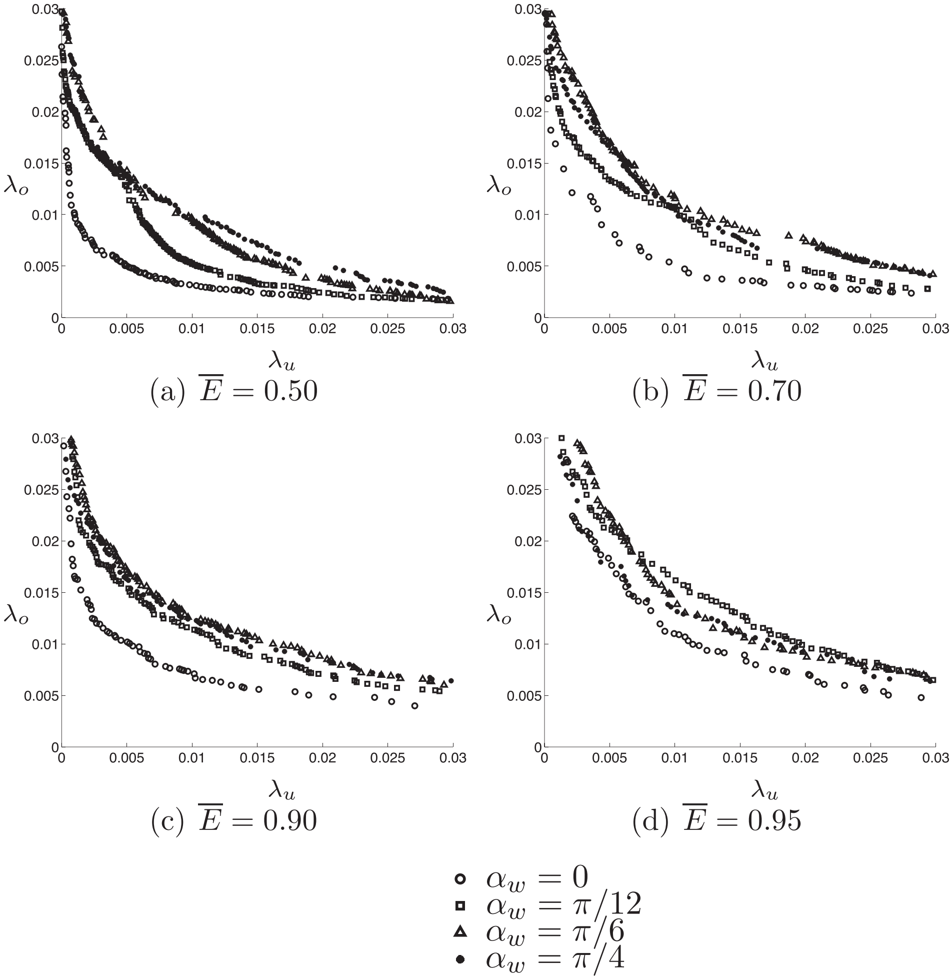
4.4. Fire shapes with the proposed RBF correction under homogeneous conditions

The fire shapes obtained with the proposed approach (CA with spread-velocity corrected with the RBF model) for different values of ellipse eccentricity (0.5, 0.7, 0.9 and 0.95) are shown in Figs. 5–8, respectively, for different wind angles. The corresponding values for error factors λ_s, λ_u and λ_o are summarized in Table 9. The benefits of the proposed approach are evident, both graphically (through the similarity between expected and predicted fire shapes) and numerically. The values obtained for the errors factors are lower than the ones obtained with the single-objective approach: this proves both the advantage of using a multi-objective formulation for the optimization problem and the effectiveness of the RBF methodology for approximating the optimal corrections during run-time. Considering the largest eccentricity case with wind in the direction of one of the grid axes, the reduction in shape factor's value is 93.9% relative to the 24-cell neighborhood RCA and 70.8% relative to the 24-cell neighborhood ECA taken as reference. Thanks to the bi-objective optimization, the error is equally divided between under- and over-predicted areas. Relatively to the reference algorithms, the under-prediction factor λ_u is reduced by 97.1% (RCA) and 80% (ECA). The over-prediction factor λ_o has also been reduced (−56%) relative to the one found in the 24-cell neighborhood ECA (the RCA has a zero value for λ_o). The influence of the effective wind direction on the fire shape is also minimal (9).

Furthermore, in comparison to the 8-cell versions of the reference solvers, improvements are even more significant. Using a 8-cell neighborhood in place of the 24-cell neighborhoods used by the reference solvers provides several advantages, such as the reduced complexity of the algorithm (shorter computational times), reduced memory requirements and better accuracy in the presence of rapidly changing landscapes.

Even if a comparison of the computational times required by different numerical algorithms is not straight-forward (they are highly dependent on the specific implementations), the potential for low running time of cell-based approaches is widely recognized, thanks to the avoidance of the de-looping algorithm necessary to reconstruct the fire perimeter at the end of each time step in vector-based implementations. Other advantages of raster-based algorithms are their computational agility and portability to parallel computing environment.

A quantification of the additional computational cost of the proposed algorithm (relative to the standard CA) is not immediate, either, as it depends on the specific implementation: the time scales of the properties used in a fire-spread simulation vary significantly – terrain slope, fuel load, depth, surface-to-volume ratio, heat content can be considered constant in the time-frame of a simulation, others (such as fuel moisture) change slowly, while other can change more rapidly (wind speed and direction). The computational overhead of the proposed approach is largely dependent on the frequency used to update the above quantities (which determines



(a) $\overline{E} = 0.50$ (b) $\overline{E} = 0.70$

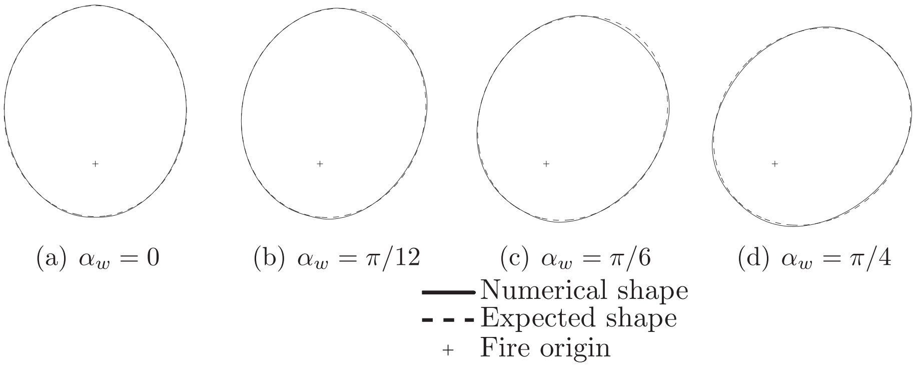
(c) $\overline{E} = 0.90$ (d) $\overline{E} = 0.95$

Fig. 4. Pareto fronts from the multi-objective optimization.

how often the outputs of Table 2 need to be recalculated). In the worst case scenario (i.e. the above quantities are calculated only once at the beginning of the simulation) the increase in computational time due to the proposed modification is of 10.8% (this was calculated for the highest eccentricity simulation, subtracting the computational times required for initialization and i/o operations, using one core of a dual quad-core 2.53 GHz Intel Xeon processor).

4.5. Comparison with Farsite on real topographies

In order to verify its performance on a realistic spread problem, the proposed algorithm was compared to a vector-based simulator (Farsite (Finney, 2004)) on a real-surface topography under different wind conditions. Only surface fires have been considered, and spotting and acceleration modules were disabled in Farsite, as



(a) $\alpha_w = 0$ (b) $\alpha_w = \pi/12$ (c) $\alpha_w = \pi/6$ (d) $\alpha_w = \pi/4$

Fig. 5. Comparison of expected and numerical two-dimensional fire-spread shapes under homogeneous conditions, for different wind angles ($\overline{E} = 0.50$).

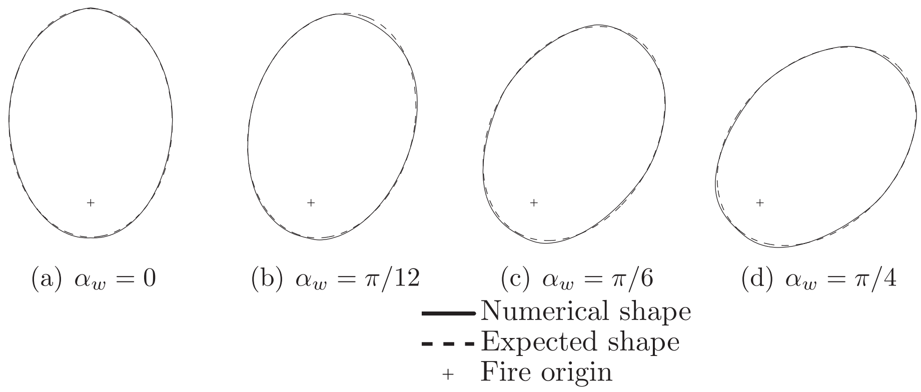

(a) $\alpha_w = 0$ (b) $\alpha_w = \pi/12$ (c) $\alpha_w = \pi/6$ (d) $\alpha_w = \pi/4$

Fig. 6. Comparison of expected and numerical two-dimensional fire-spread shapes under homogeneous conditions, for different wind angles ($\overline{E} = 0.70$).

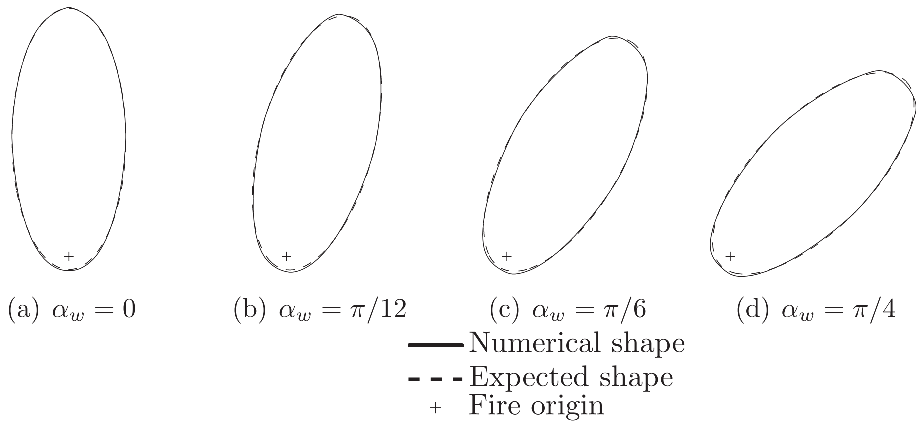

(a) $\alpha_w = 0$ (b) $\alpha_w = \pi/12$ (c) $\alpha_w = \pi/6$ (d) $\alpha_w = \pi/4$

Fig. 7. Comparison of expected and numerical two-dimensional fire-spread shapes under homogeneous conditions, for different wind angles ($\overline{E} = 0.90$).

the purpose of this exercise was to evaluate the performance of the proposed fire-spread approach. The numerical grid used by the CA solver is constituted by 1.44 million square cells with 10 m sides. Farsite does not make use of a fixed numerical grid, but discretizes the fire perimeter directly. The maximum distance between adjacent computational points and their maximum displacement has been set to values similar to the ones used by the CA solver. A uniform fuel bed corresponding to the standard fuel model 2 (grass) (Anderson et al., 1982) was used.

Three scenarios were simulated: a zero-wind situation, a scenario with a 10 mph domain-averaged wind and one with a 20 mph domain-averaged wind (measured at 20 feet agl), directed from West to East. The three scenarios are labeled T0, T10 and T20, for simplicity. Gridded wind vectors were obtained through a mass-consistent approach (Forthofer, 2007).

According to Farsite, the surface fire burned a total of 5.96 km^2 in 12 h in T0, 12.07 km^2 in 4 h in T1 and 7.08 km^2 in 1 h in T2. Figs. 9–11 show a comparison of the fire shaped predicted by

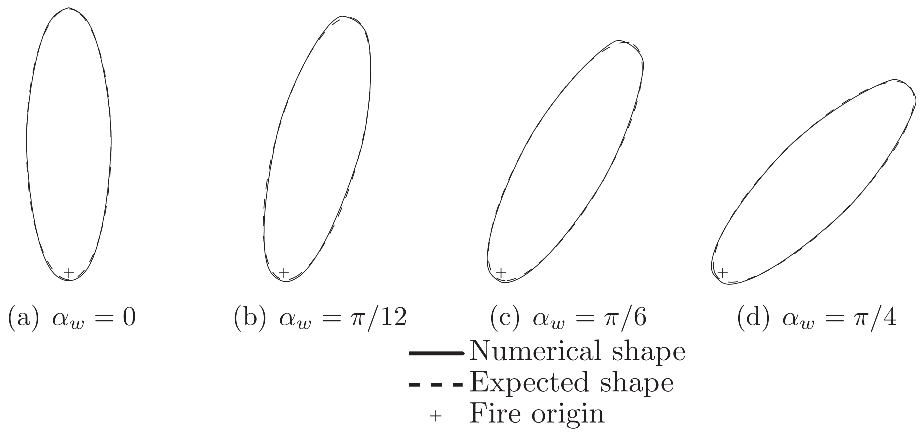
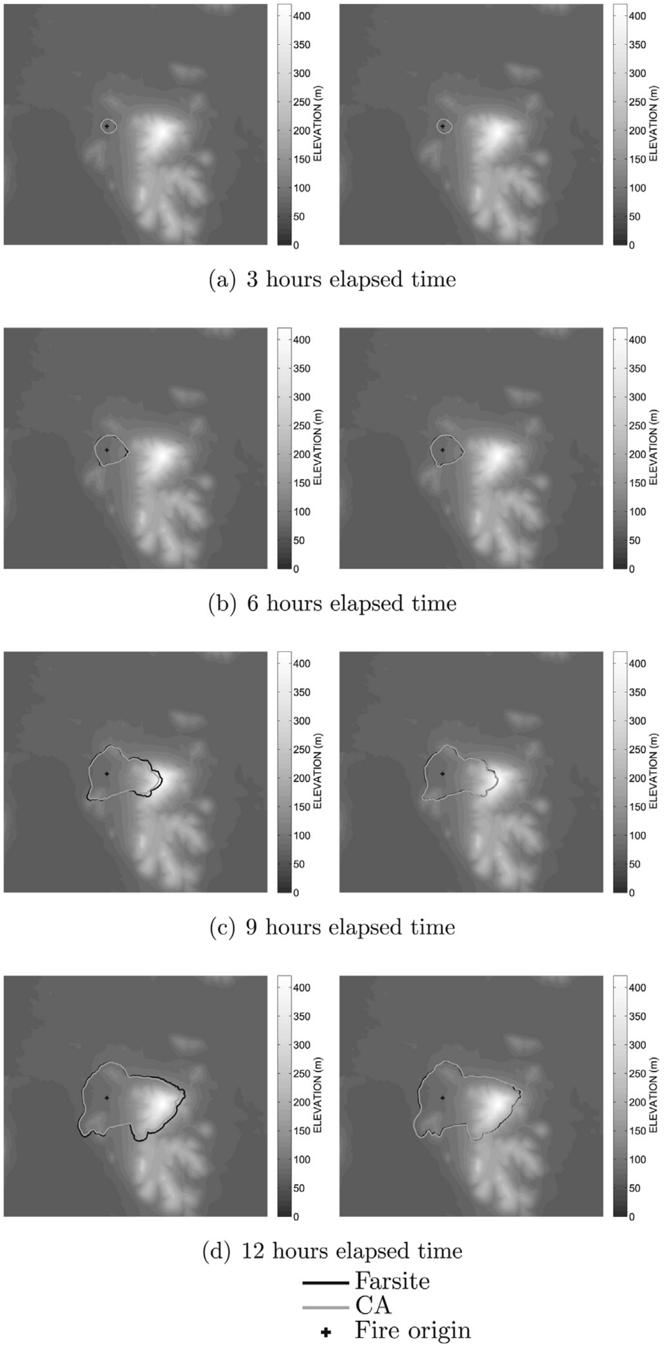

(a) $\alpha_w = 0$ (b) $\alpha_w = \pi/12$ (c) $\alpha_w = \pi/6$ (d) $\alpha_w = \pi/4$

Fig. 8. Comparison of expected and numerical two-dimensional fire-spread shapes under homogeneous conditions, for different wind angles ($\overline{E} = 0.95$).

Table 9
Dependence of error factors λ_s, λ_u and λ_o on the wind angle α_w (two-objective optimization results).

$\overline{E}$	0.50				0.70			
α_w	0	$\pi/12$	$\pi/6$	$\pi/4$	0	$\pi/12$	$\pi/6$	$\pi/4$
λ_s	0.010	0.014	0.019	0.019	0.014	0.020	0.021	0.020
λ_u	0.005	0.007	0.010	0.010	0.007	0.010	0.011	0.010
λ_o	0.005	0.007	0.010	0.010	0.007	0.010	0.010	0.010
$\overline{E}$	0.90				0.95			
α_w	0	$\pi/12$	$\pi/6$	$\pi/4$	0	$\pi/12$	$\pi/6$	$\pi/4$
λ_s	0.016	0.022	0.023	0.023	0.021	0.027	0.024	0.025
λ_u	0.008	0.011	0.011	0.011	0.010	0.014	0.012	0.013
λ_o	0.008	0.011	0.012	0.012	0.011	0.014	0.012	0.012



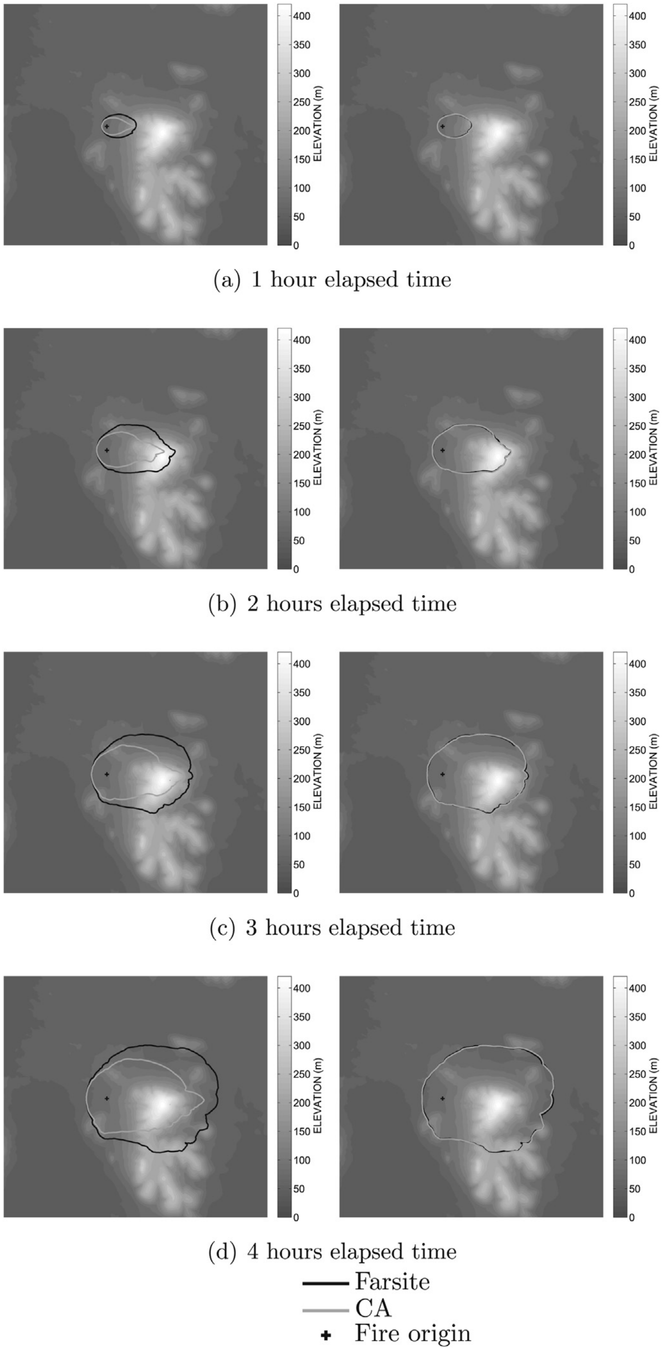
(a) 3 hours elapsed time

(b) 6 hours elapsed time

(c) 9 hours elapsed time

(d) 12 hours elapsed time

Fig. 9. Comparison of fire perimeters predicted by Farsite and the CA (zero-wind conditions), without (left) and with the proposed correction (right).



(a) 1 hour elapsed time

(b) 2 hours elapsed time

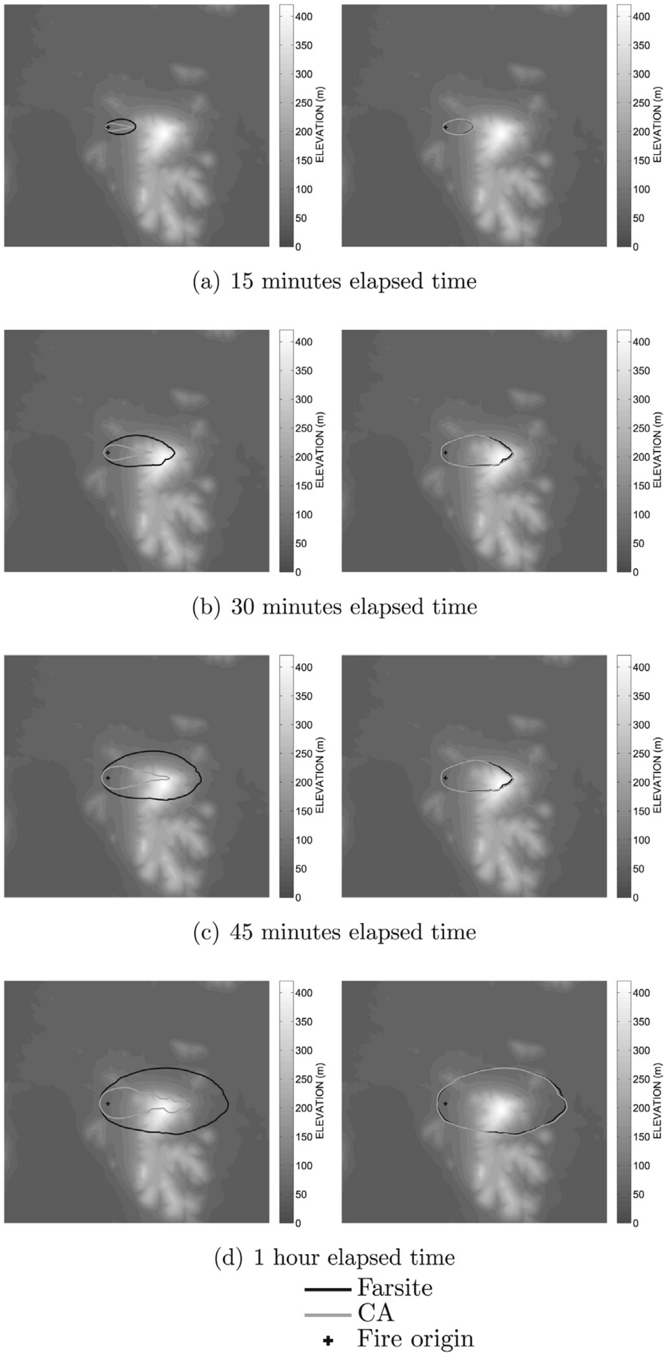
(c) 3 hours elapsed time

(d) 4 hours elapsed time

Fig. 10. Comparison of fire perimeters predicted by Farsite and the CA solver (10 mph average wind conditions), without (left) and with the proposed correction (right).

Farsite and the CA solver, both without (left) and with the proposed correction, for different elapsed times. The agreement is evident.

Tables 10–12 report the shape disagreement factors for the three experiments, at four intermediate times. The values are significantly reduced. Table 13 compares the wall clock times: for complex simulation (the combined presence of variable slope and wind), the reduction in wall clock time can be in excess of 75%. The low values obtained for the errors factors prove both the success of the optimization approach for finding the optimal values for the correction factors and the effectiveness of the RBF methodology for approximating the optimal corrections during run-time (i.e. for conditions different from the ones the corrections factors were optimized for).



(a) 15 minutes elapsed time

(b) 30 minutes elapsed time

(c) 45 minutes elapsed time

(d) 1 hour elapsed time

Fig. 11. Comparison of fire perimeters predicted by Farsite and the CA solver (20 mph average wind conditions), without (left) and with the proposed correction (right).

These results have been obtained with 8-cell neighborhoods, while several other raster-based solvers make use of 24-cell (or larger) neighborhoods to mitigate the distortion of fire perimeters, thus increasing algorithm complexity and memory requirements and reducing the accuracy in the presence of rapidly changing landscapes. The solver is implemented in *C* and compiled with the open-source gcc 4.7.2 compiler with -O2 optimization under Linux environment. All simulations have been performed on a single-core of a dual quad-core 2.53 GHz Intel Xeon processor, to allow comparison with Farsite (a vector based simulator) on an equal basis. Raster based simulators are better suited to parallel computing environment, where the proposed approach should provide even larger computational gains.

Table 10
Error factors λ_s, λ_u and λ_o for the T0 experiment.

Elapsed time	Without correction			With correction		
	λ_s	λ_u	λ_o	λ_s	λ_u	λ_o
3 h	0.070	0.028	0.044	0.063	0.029	0.035
6 h	0.064	0.028	0.038	0.062	0.025	0.039
9 h	0.095	0.093	0.003	0.029	0.018	0.011
12 h	0.090	0.088	0.002	0.029	0.010	0.019

Table 11
Error factors λ_s, λ_u and λ_o for the T1 experiment.

Elapsed time	Without correction			With correction		
	λ_s	λ_u	λ_o	λ_s	λ_u	λ_o
1 h	0.457	0.457	0.000	0.046	0.010	0.036
2 h	0.457	0.457	0.000	0.023	0.007	0.016
3 h	0.425	0.425	0.000	0.019	0.008	0.014
4 h	0.436	0.436	0.000	0.018	0.007	0.011

Table 12
Error factors λ_s, λ_u and λ_o for the T2 experiment.

Elapsed time	Without correction			With correction		
	λ_s	λ_u	λ_o	λ_s	λ_u	λ_o
15 min	0.646	0.646	0.000	0.047	0.007	0.040
30 min	0.730	0.730	0.000	0.044	0.021	0.024
45 min	0.747	0.747	0.000	0.031	0.016	0.015
1 h	0.738	0.738	0.000	0.026	0.008	0.018

Table 13
Comparison of Wall Clock Times between Farsite and the CA approach.

Experiment	Farsite	OCA
T0	979 s	408 s
T1	1086 s	283 s
T2	550 s	128 s

5. Conclusions

This work presents a Cellular Automata approach for wildland fire prediction that makes use of a modified advection speed in the differential equation describing the fire-front behavior to achieve fire perimeters resembling the expected elliptic fire shapes more closely. The values for the correction factors used in this method are optimized through a numerical optimization approach, which makes use of a well-known algorithm for multi-objective optimization problems.

The proposed approach obtains fire perimeters closely matching the results of vector-based simulators, with all the advantages of raster-based techniques (simpler implementation, better portability to parallel computing environments), while making use of 8-cell neighborhoods, which better accuracy in the presence of rapidly changing landscapes (fuel, topography, wind).

Future work includes testing in real-world scenarios (including comparison to available experimental data), parallelization to both CPU and GPU architectures and implementation within CFD solvers able to simulate the two-way interaction between fire and wind.

References

Alexander, M.E., 1985. Estimating the length-to-breadth ratio of elliptical forest fire patterns. In: Proceedings of the 8th Conference on Fire and Forest Meteorology, pp. 287–304.

Anderson, D.H., Catchpole, E., De Mestre, N.J., Parkes, T., 1982. Modeling the spread of grass fires. J. Aust. Math. Soc. Ser. B Appl. Math. 23 (04), 451–466.

Ball, G.L.G., Guertin, D.P., 1992. Improved fire growth modeling. Int. J. Wildland Fire 2 (2), 47–54.

Belvins, C.D., Apr 2015. Firelib User Manual and Technical Reference. URL. http://www.fire/firelib/doc.html.

Bennett, N., Croke, B., Guariso, G., Guillaume, J., Hamilton, S., Jakeman, A., Marsili-Libelli, S., Newham, L., Norton, J., Perrin, C., Pierce, S., Robson, B., Seppelt, R., Voinov, A., Fath, B., Andreassian, V., 2013. Characterising performance of environmental models. Environ. Model. Softw. 40, 1–20. http://dx.doi.org/10.1016/j.envsoft.2012.09.011.

Bogdos, N., Manolakos, E., 2013. A tool for simulation and geo-animation of wildfires with fuel editing and hotspot monitoring capabilities. Environ. Model. Softw. 46, 182–195. http://dx.doi.org/10.1016/j.envsoft.2013.03.009.

Cencerrado, A., Corts, A., Margalef, T., 2014. Response time assessment in forest fire spread simulation: an integrated methodology for efficient exploitation of available prediction time. Environ. Model. Softw. 54, 153–164. http://dx.doi.org/10.1016/j.envsoft.2014.01.008.

Du, Y., Chen, D., Xiang, X., Tian, Q., Zhang, Y., 2013. Topological design of structures using a cellular automata method. CMES Comput. Model. Eng. Sci. 94 (1), 53–75.

Duff, T., Chong, D., Tolhurst, K., 2013. Quantifying spatio-temporal differences between fire shapes: estimating fire travel paths for the improvement of dynamic spread models. Environ. Model. Softw. 46, 33–43. http://dx.doi.org/10.1016/j.envsoft.2013.02.005.

D'Ambrosio, D., Di Gregorio, S., Gabriele, S., Gaudio, R., 2001. A cellular automata model for soil erosion by water. Phys. Chem. Earth Part B Hydrol. Oceans Atmos. 26 (1), 33–39.

D'Ambrosio, D., Iovine, G., Spataro, W., Miyamoto, H., 2007. A macroscopic collisional model for debris-flows simulation. Environ. Model. Softw. 22 (10), 1417–1436.

Finney, M.A., 2002. Fire growth using minimum travel time methods. Sci. N. Y. 1420–1424. http://dx.doi.org/10.1139/X02-068.

Finney, M.A., 2004. FARSITE: Fire Area Simulator-model Development and Evaluation. Tech. Rep. RMRS-RP-4. USDA, Ogden, UT.

Finney, M.A., 2006. An overview of FlamMap Fire Modeling capabilities. No. RMRS-P-41. In: Fuels Management – How to Measure Success. USDA Forest Service.

Fonseca, C.M., Fleming, P.J., 1995. An overview of evolutionary algorithms in multiobjective optimization. Evol. Comput. 3, 1–16.

Forthofer, J.M., 2007. Modeling Wind in Complex Terrain for Use in Fire Spread Predictions (Ph.D. thesis).

Frandsen, W.H., Andrews, P.L., 1979. Fire Behavior in Non-uniform Fuels. Tech. rep.. USDA, Ogden, UT.

French, I.A., Anderson, D., Catchpole, E.A., 1990. Graphical simulation of bushfire spread. Math. Comput. Model. 13 (12), 67–71.

Ghisu, T., Parks, G.T., Jarrett, J.P., Clarkson, P.J., 2010. Adaptive polynomial chaos for gas turbine compression systems performance analysis. AIAA J. 48 (6), 1156–1170. http://dx.doi.org/10.2514/1.J050012.

Ghisu, T., Parks, G.T., Jaeggi, D.M., Jarrett, J.P., Clarkson, P.J., 2010. The benefits of adaptive parametrization in multi-objective Tabu Search optimization. Eng. Optim. 42 (10), 959–981. http://dx.doi.org/10.1080/03052150903564882.

Ghisu, T., Parks, G.T., Jarrett, J.P., Clarkson, P.J., 2011. An integrated system for the aerodynamic design of compression systems. Part I: development. J. Turbomach. 133 (1), 011011. http://dx.doi.org/10.1115/1.4000534.

Ghisu, T., Parks, G.T., Jarrett, J.P., Clarkson, P.J., 2011. An integrated system for the aerodynamic design of compression systems. Part II: application. J. Turbomach. 133 (1), 011012. http://dx.doi.org/10.1115/1.4000535.

Ghisu, T., Parks, G.T., Jarrett, J.P., Clarkson, P.J., 2011. Robust design optimization of gas turbine compression systems. J. Propuls. Power 27 (2), 282–295. http://dx.doi.org/10.2514/1.48965.

Glasa, J., Halada, L., 2008. On elliptical model for forest fire spread modeling and simulation. Math. Comput. Simul. 78 (1), 76–88. http://dx.doi.org/10.1016/j.matcom.2007.06.001.

Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publishers, USA.

Golberg, M.A., Cho, H.A., 2004. Introduction to Regression Analysis. WIT Press.

Green, D.G., Gill, A.M., Noble, I.R., 1983. Fire shapes and the adequacy of fire-spread models. Ecol. Model. 20 (1), 33–45.

Gregorio, S.D., Filippone, G., Spataro, W., Trunfio, G.A., 2013. Accelerating wildfire susceptibility mapping through GPGPU. J. Parallel Distrib. Comput. 73 (8), 1183–1194.

Hernández Encinas, L., Hoya White, S., Martín del Rey, A., Rodríguez Sánchez, G., 2007. Modelling forest fire spread using hexagonal cellular automata. Appl. Math. Model. 31 (6), 1213–1227. http://dx.doi.org/10.1016/j.apm.2006.04.001.

Hooke, R., Jeeves, T.A., 1961. 'Direct search' solution of numerical and statistical problems. J. Assoc. Comput. Mach. 2 (8), 212–229.

Jaeggi, D.M., Parks, G.T., Kipouros, T., Clarkson, P.J., 2008. The development of a multi-objective Tabu Search algorithm for continuous optimisation problems. Eur. J. Oper. Res. 185, 1192–1212.

Johnston, P., Kelso, J., Milne, G.J., 2008. Efficient simulation of wildfire spread on an irregular grid. Int. J. Wildland Fire 17 (5), 614–627. http://dx.doi.org/10.1071/WF06147.

Karafyllidis, I., Thanailakis, A., 1997. A model for predicting forest fire spreading using cellular automata. Ecol. Model. 99 (1), 87–97.

Lopes, A.M.G., Cruz, M.G., Viegas, D.X., 2002. FireStation – an integrated software system for the numerical simulation of fire spread on complex topography. Environ. Model. Softw. 17 (3), 269–285. http://dx.doi.org/10.1016/S1364-8152(01)00072-X.

McArthur, A.G., 1966. Weather and Grassland Fire Behaviour. Tech. Rep. Leaflet 100. Commonwealth Department of National Development, Forestry and Timber Bureau, Canberra, ACT.

Mell, W., Jenkins, M.A., Gould, J., Cheney, P., 2007. A physics-based approach to modelling grassland fires. Int. J. Wildland Fire 16 (1), 1–22. http://dx.doi.org/10.1071/WF06002.

Perry, G.L.W., 1998. Current approaches to modelling the spread of wildland fire: a review. Prog. Phys. Geogr. 22 (2), 222–245. http://dx.doi.org/10.1177/030913339802200204.

Peterson, S.H., Morais, M.E., Calson, J.M., Dennison, P.E., Roberts, D.A., Moritz, M.A., Weise, D.R., 2009. Using HFIRE for Spatial Modeling of Fire in Shrublands. Tech. Rep. PSW-RP-259. USDA, Albany, CA.

Richards, G.D., 1995. A general mathematical framework for modeling two-dimensional wildland fire spread. Int. J. Wildland Fire 5 (2), 63–72.

Rongo, R., Spataro, W., D'Ambrosio, D., Vittoria Avolio, M., Trunfio, G., Di Gregorio, S., 2008. Lava flow hazard evaluation through cellular automata and genetic algorithms: an application to Mt Etna volcano. Fundam. Inf. 87 (2), 247–267.

Rothermel, R.C., 1972. A Mathematical Model for Predicting Fire Spread in Wildland Fuels. Tech. Rep. INT-115. USDA, Ogden, UT.

Rothermel, R.C., 1983. How to Predict the Spread and Intensity of Forest and Range Fires. Tech. Rep. INT-143. USDA, Ogden, UT.

Salles, T., Lopez, S., Cacas, M., Mulder, T., 2007. Cellular automata model of density currents. Geomorphology 88 (1–2), 1–20.

Sousa, S.A., dos Reis, R.J.N., Pereira, J., 2012. Simulation of surface fire fronts using fireLib and GPUs. Environ. Model. Softw. 38, 167–177. http://dx.doi.org/10.1016/j.envsoft.2012.06.006.

Sullivan, A.L., 2009. Wildland surface fire spread modelling, 1990–2007. 1: Physical and quasi-physical models. Int. J. Wildland Fire 18 (4), 349–368. http://dx.doi.org/10.1071/WF06143.

Sullivan, A.L., 2009. Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models. Int. J. Wildland Fire 18 (4), 369–386. http://dx.doi.org/10.1071/WF06142.

Trunfio, G.A., D'Ambrosio, D., Rongo, R., Spataro, W., Di Gregorio, S., 2011. A new algorithm for simulating wildfire spread through cellular automata. ACM Trans. Model. Comput. Simul. 22 (1), 1–26. http://dx.doi.org/10.1145/2043635.2043641.

Van Wagner, C.E., 1998. Modelling Logic and the Canadian Forest Fire Prediction System. For. Chron. 74 (1), 50–52.

Zakhama, R., Abdalla, M., Smaoui, H., Grdal, Z., 2009. Multigrid implementation of cellular automata for topology optimization of continuum structures. CMES Comput. Model. Eng. Sci. 51 (1), 1–24.

Zhou, Y., Mi, C., 2012. Modeling train movement for moving-block railway network using cellular automata. CMES Comput. Model. Eng. Sci. 83 (1), 1–21.

Zhou, Y., Yang, X., Mi, C., 2013. Model predictive control for high-speed train with automatic trajectory configuration and tractive force optimization. CMES Comput. Model. Eng. Sci. 90 (6), 415–437.

Zhu, Z., Liu, C., 2000. Simulation of anisotropic crystalline etching using a continuous cellular automata algorithm. CMES Comput. Model. Eng. Sci. 1 (1), 11–19.