33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ENVIRONMENTAL AGENT-BASED MODELLING FOR A FIREFIGHTING SYSTEM OF SYSTEMS

Jorge L. Lovaco[1], Ingo Staack[1] & Petter Krus[1]

[1]Department of Management and Engineering (IEI), Linköpings university, Sweden

Abstract

In the field of System-of-Systems (SoS) engineering, the study of interactions between complex systems from a holistic point of view is important for finding emerging behaviours. To observe as many behaviours as possible, especially when field testing is not a viable option, simulations play an important role in design space exploration and formulation for the Firefighting SoS framework. The presented work describes an Agent-Based Model (ABM) approach for simulation of wildfire spread and its detection using collaborating vehicles: Unmanned Aerial Vehicles (UAVs), or partly autonomous air- and land-based vehicles. Implemented in the open-source software NetLogo, the usage of its Geographic Information System (GIS) extension allows to simulate scenarios at specific locations. This ABM will be used in the future for Agent-Based Simulations (ABS) for the study of an SoS framework oriented to firefighting, including design and optimization of the SoS, constituent systems and their subsystems.

Keywords: Systems of Systems, Agent-Based Simulation, Aerospace Systems, Cyber-Physical Modelling, Wildfire Detection

1. Introduction

Wildfire prevention is becoming more challenging worldwide due to problems such as increasing temperatures, lack of rain or human related activities, especially during summer season, hence early detection is crucial [5]. Individual country-specific approaches are taken for prevention, detection, or extinction, but differences in vegetation, geography, settlement, resources, and infrastructure may not allow a standardised solution. An study of the Firefighting System-of-Systems *(SoS)* framework is introduced here with the aim of finding an optimized solution for the Swedish territory.

1.1 High-performance Wildfire Simulation Model

The Firefighting *SoS* framework is studied to discuss what it could bring to wildfire detection and fighting for optimization of resources, which involves: the systems to be used for assembling the *SoS*, improved tactics such as shorter flight paths, aircraft and their subsystems design and refinement, etc. To do so, an Agent-Based Model (*ABM*) to be used for performing Agent-Based Simulations (*ABS*) is presented here.

1.2 The Need for a Model

For an effective Firefighting *SoS* study, it is necessary to model an environment with a behaviour similar to the actual. As affirmed by Ross Ashby ([18]): *"the regulation of a system is only efficient if it leans on a control system as complex as the system that is controlled".* Depending on the level of centralization needed or desired, it will be necessary to develop a complex control for it. The same train of thoughts used for expanding the concept of controlling a system to controlling an *SoS* can be used again for system engineering (*SE*).
The concept of *SoSE* can be seen as *SE* at *SoS* level. This allows a double way engineering process: on one hand, it is possible to use the models built to simulate with the systems currently available

and compare different *SoS* resulting values; on the other hand, it is possible to simulate an *SoS* conformed by systems with a set of desired characteristics and reverse engineer these systems to define a design space from the results. In this sense, *SoS* and *SoSE* can form an iterative design and optimization loop.

1.3 Additional System of Systems Identified Needs

Being an *SoS* an assembly of systems, combinatory becomes a part of the *SoS* analysis, hence becoming necessary to sustain *traceability*. Through traceability it is possible to propagate changes among different levels [17], namely concepts, requirements, specifications, decisions or impact studies. Data analysis and visual analytics are examples of disciplines that can ease comparison of the different resultant combinations during the study.

There is also a need for *standards* in any *SoS* framework development or study to set common boundaries, proper knowledge transfer and communication between different groups, and/or common practices that facilitate innovation by removing unexpectedness if possible. For example, for a given set of systems conforming an *SoS*, simulations can be used not only for evaluation or design of a decision-taking process, but also for an attempt on achieving standardisation at different levels such as data to be extracted or discarded.

Humans as a factor in an *SoS* and their role should be accounted for in every step. Whether as an operator or as a supervisor [18]. The need for *communication interfaces* will influence the design and usage of the different systems and therefore have a big weight in the performance and flexibility of an *SoS*.

2. A Discrete Approach: Agent-Based Simulations

For *SoS* simulation for systems deployment, it is important to start by considering the approach of a continuous or discrete approach. The latter has been taken rather than the former for wildfire simulations. Discrete simulations process the events that occur at different instants or several of them at the same time. This enhances the simulation speeds since, when no event is triggered, even with small time steps, the computational power needed becomes very small. The discrete approach also allows to avoid very complex governing equations based on differential equations. For these reasons, the discrete-event approach is preferred, focusing on high-speed and easy to understand simulations.

2.1 Relevancy of agent-based simulations

By using an *ABM*, it is possible to take individual computational entities named agents, with each of them having a collection of attributes and spatial location [7]. Attributes and location can be parameters or variables dependant of the time-wise evolution of external inputs from the environment or interactions with other agents. The number of agents can be static or variable as well. They can not only be different in their attributes, but also in their *species*, having each species behaviours based on own rules. Classic examples are self-regulating, dynamic (biological) systems such as the sheep and wolf models, where sheep will eat grass to survive and reproduce whilst wolves will survive and reproduce by hunting the sheep [28]. Frequently, *ABS* are used to model or find/detect emergent behaviours from interactions between agents influenced by an environment. Moreover, *ABS "...offer insight into the adaptation and function of systems over time and space when access to field data is dangerous, unavailable, or simply impossible to collect"* [19]. Recent examples of successful *ABS*s are [3], [8] or [16]. By having individual vegetation agents in the model, it becomes possible to define individual parameters, measure local distances, local densities, etc., and be influenced by localized changes, for instance, in temperature or humidity. It also permits to import the vegetation that has been accounted for in an area, adding value to the model from a holistic point of view. For the case of smoke plumes, treating them as collections of particles allows to move them individually depending on their variables and stochasticity, enabling to use simplified models and to avoid the more complex forms of continuum fluids and their dynamics.

3. Implementation

The *ABS* for this project has been realized using NetLogo [27]. The choice was based on its functionalities [25] and capabilities for modelling several thousands of independent agents [19] with their own set of characteristics variables, parameters and rules defining their behaviour [23].

3.1 Ambient conditions

The environment parameters need to be defined when setting up the simulation. These parameters cannot be modified by the model but can be changed by the user at any moment. For example, to modify the simulation accordingly to weather station data or to study an *SoS* with dynamic environmental conditions. The defined environmental parameters are shown in Table 1

Table 1 – List of environmental parameters

Parameter	Definition	Value range	Units
Day/night-time	User	Day / Night	-
Temperature	User	-20 to +45	[Celsius]
Wind velocity	User	0 to 90	[km/h]
Wind angle	User	0 to 360	[degrees]
Relative humidity	User	10 to 90	-

3.2 Patches

For *ABS*, patches represent the (squared) grid of the model. The Geographic Information System (*GIS*) elevation data, imported from [12] to the model, has a resolution of 100 meters, corresponding each patch to an area of 100 m x 100 m by default. The patch dimensions are relevant because the sensor models in the simulation compute radial distances based on patch centroid coordinates and thus, depending on the resources available and accuracy needed, it is possible to modify the resolution by changing the number of patches in the grid and/or the number of data points. A variable named *smoke-smell* has been declared for each patch and is changed by smoke agents in the proximity.

3.3 Agents

The collection of agents are trees (vegetation), smoke particles and vehicles. Note that for the study at hand, only aerial vehicles will be used in the simulations, but it is possible to add other vehicle types as well. The agent characteristics are summarized in Table 2.

Table 2 – Agent characteristics.

Agents	Position	Behaviour
Trees (vegetation)	Fixed	Complex
Smoke particles	Varying (passive)	Simple
Airborne vehicles	Varying (active)	Complex

3.3.1 Tree Agents

Each tree agent represents one specific type of plant or tree. To adjust the agent to the vegetation in the simulated area, the following agent parameters and variables in Table 3 (adapted from Fons [10]) must be set:
The tree agents variables represent respectively:

- Self temperature: Characteristic variable of each agent storing their temperature at every time instant. Initial value taken from the ambient temperature parameter.

Table 3 – Fire forward spread parameters. Adapted from [4]

Parameter name	Definition	Default Value	Units
Ignition temperature	User	290	Celsius
Flame temperature	User	820	Celsius
Variable name	**Definition**	**Range**	**Units**
Self-temperature	Computed	[Ambient, Flame temperature]	Celsius
Self-material	Randomised	[500, 1000]	-
Heading angle	Computed	[0, 360]	Degrees

- Self material: Characteristic variable of each agent storing their amount of burnable material (randomised or used input).
- Heading angle: Characteristic variable of each agent storing their forward spread angle computed at each instant in time as the arctangent of the resultant vector addition of the wind velocity and local slope vectors [23].

Each tree agent comprises of the behavioural rules in Table 4:

Table 4 – Behavioural rules for tree agents

Rules	Description
Temperature increase	Tree agents on fire increase the temperature of their surroundings. The increment depends on the position, hence based on the forward, flanking or backing spreads.
Catch fire	Tree agents with a temperature equal or higher than the ignition temperature will catch on fire, changing their shape and colour from "tree" and "green" to "fire" and "red" respectively.
Temperature saturation	Tree agents with a temperature equal or beyond the flame temperature will not raise their temperature anymore. This prevents nonphysical values of temperature and numerical errors due to values out of limits for long simulations.
Colouring	Tree agents with self-material equal to 0 change their shape and colour to "tree" and "black" respectively.
Smoke emission	Tree agents on fire "hatch" or generate a smoke agent every time step they are on fire.

Fire spread models are challenging to develop due to all the involved physics and uncertainties. A classic reference is the work by Fons [10] from 1946, which was used, among others, by Anderson [4] for representing fire spreads under wind conditions with the help of a double ellipse model. The resulting shapes from the double ellipse model is shown in Figure 1. Local geography, however, involves changes in the local slope and aspect, leading to wind-slope interactions for the local fire spread rate. Adopting the study done by [23], influences on the fire spread rate from both local terrain slope and wind are included in the simulation model.

Following a procedure similar to [10], and to the possible extent using the same notation, the model is transformed for computations of the temperature increments for discrete time steps. Under the assumption that a fuel particle is represented by an agent in the ABM, a rate of fire spread R, a

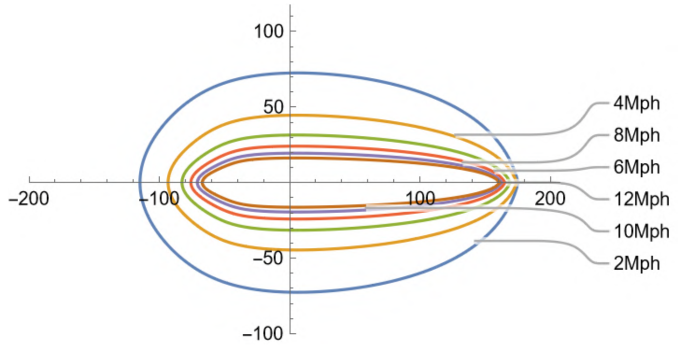


Figure 1 – Double ellipse fire spread model under different wind conditions [4].

distance *L* between particles and an ignition time between particles θ_i, it follows:

$$R = \frac{L}{\theta_i} \tag{1}$$

Starting with the left-hand side of the Equation 1, the rate of spread *R* is assumed as a function of terrain slope, wind velocity, fuel moisture content and vegetation characteristics.

Table 5 – Fire forward spread. Adapted from Anderson [4]

Wind velocity [km/h]	Fuel moisture content [%]	*R*-Forward spread [m/s]
0.00	4	0.0049
3.22	4	0.0109
6.44	4	0.0198
9.66	4	0.0335
12.88	4	0.0525
16.1	4	0.0777
19.32	4	0.1099

With the fire spread values shown in Table 5(data adapted from [4]), *R* was approximated to a non-linear over the wind velocity, w, up to values of 20 kilometres per hour. Due to the lack of data for stronger winds, a 10% of the wind velocity rule of thumb is taken as suggested in [6], resulting in a piece wise function shown in Figure 2:

$$R_w[m/s] = \begin{cases} -0.0152693 + e^{(-3.9461+0.0969421w)} & : w \leq 20[km/h] \\ w * 0.1/3.6 & : 20 < w[km/h] \end{cases} \tag{2}$$

The relative humidity (*RH*) value can be used to approximate the fuel moisture content (*FMC*). With data from [22], the fitted nonlinear model using [12] results in:

$$FMC = 2.86682 + 0.00634132 * RH^{1.74816} \tag{3}$$

The spread rate model with the wind velocity as variable is fitted again to another non-linear model [12] with *FMC* as another variable.

$$R_{m,w}[m/s] = -0.000228 + (2.588376 * (RSR * (R_w)))/FMC^{0.671335} \tag{4}$$

Note that *FMC* is a dimensionless variable and the relative spread rate (*RSR*) a correction parameter that depends on the vegetation characteristics. To be able to model the effect of the terrain slope in the spread of a fire, the wind-slope correction by McArthur from [23] is used. This results in the final value of *R* as a function of the moisture content, wind velocity and slope:

$$R(m, w, \gamma_S)[m/s] = R_{m,w} e^{0.069\gamma_S} \tag{5}$$

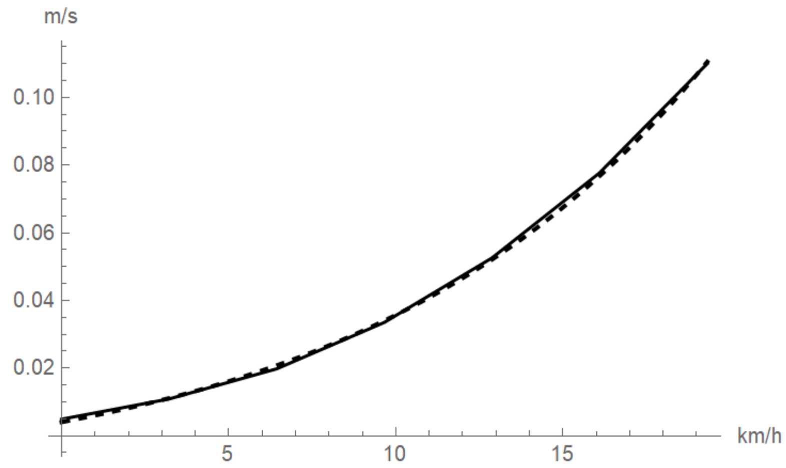


Figure 2 – Comparison of data (solid line) and Equation 2 (dashed line).

Note that for flanking spreads it is assumed zero slope whilst for the backing spread, the value of the local slope is negative. Flanking and backing spread corrections are respectively [4]:

$$R_b[m/s] = R_{m,w} * 0.534e^{-0.1147w} \tag{6}$$

$$R_c[m/s] = R_{m,w} * 0.492e^{-0.1845w} \tag{7}$$

Considering the slope corrections, they respectively become:

$$R_\beta(m,w,\gamma_S)[m/s] = R_b e^{0.069\gamma_S} \tag{8}$$

$$R_\kappa(m,w,\gamma_S)[m/s] = R_c e^{0.069\gamma_S} \tag{9}$$

Continuing with the right-hand side of Equation 1. The ignition time, θ_i, is defined in [10] as *"...the time for the n particle to burst into flame after the (n-1) particle has ignited"* . Expressing the ignition time as L divided by R:

$$\theta_i[s] = \frac{L[m]}{R[m/s]} \tag{10}$$

To define the ignition lag as the increase of temperature divided by the time needed for the increment, being in this case, *Kelvin per seconds*, a variable Ω is introduced. Then, the variables Ω and θ_i are related as follows:

$$\Omega = \frac{\Delta T}{\theta_i} \tag{11}$$

The temperature increment, ΔT, taken here is equal to the difference between t_i and t. Following nomenclature and definitions given in [10]:

- t_i is the "ignition temperature of fuel particle", in Kelvin.
- t is the "temperature of fuel particle at time θ", in Kelvin.

Temperature units can be in Kelvin or Celsius with no difference whatsoever in the Equation 11. Bearing in mind that a fuel particle represents an agent, it is possible to continue and redefine Equation 1 in terms of Ω:

$$\theta_i = \frac{\Delta T}{\Omega} = \frac{t_i - t}{\Omega} = \frac{L}{R} \tag{12}$$

By reordering the terms in Equation 12, it is possible to express Ω as:

$$\Omega = \frac{(t_i - t)R}{L} \tag{13}$$

With Ω set, it is possible to compute the local temperature increment around each fire source for each simulation step. Also, since Ω is defined by means of the fire spread rate R, Equations 2 and 6 from [4] can be used as well for flanking and backing spread distances.

3.3.2 Smoke Agents

Another important part of the model is the smoke plume rising from the area on fire. Therefore, the model needs to be capable to reproduce smoke behaviour with some level of detail for a complete SoS study. Under daytime conditions and no weather phenomena other than wind, spotting a fire plume from an aircraft can be done visually from long distances. Under night-time conditions, with reduced visibility, other options may need to be considered. Due to the intrinsic difficulties of fluid dynamic modelling and simulation, alternative low-fidelity models tend to be favoured in the case of *ABS*, lowering computational costs. References related to smoke plume spotting are [14], [24], and [21]; whilst the main smoke plume modelling reference used in this work is [2]. The agent parameters and variables shown in Table 6 define the smoke properties, whilst the set of behavioral rules is shown in Table 7.

Table 6 – Smoke parameters.

Parameter name	**Definition**	**Default Value**	**Units**
Self-initial-x/y	Computed	[-95, 95]	-
Self-height0-smoke	Computed	Local elevation	m
Stack-gas-ejection-z-velocity	User	25	m/s
Variable name	**Definition**	**Range**	**Units**
Self-x/y	Computed	[-95, 95]	-
Self-angle-heading	Randomised	[0, 360]	Degrees
Self-height-smoke	Computed	$[0, \infty)$	m
Self-z-vel-smoke	Computed	[0, 25]	m/s

The smoke agents variables represent respectively:

- Self-x and Self-y: Coordinates of each smoke agent representing their actual position.
- Self-angle-heading: Orientation of the smoke agent in degrees.
- Self-height-smoke: Actual altitude of the smoke agent.
- Self-z-vel-smoke: Actual vertical velocity of the smoke agent.

Table 7 – Behavioural rules for smoke agents

Rules	**Description**
Acceleration	Vary velocity accordingly to the smoke plume model.
Displacement	Change position and orientation depending on the velocity.
Smell emission	Increase the smoke-smell variable of the patches in a circular area with radius 300 m.

3.3.3 Aircraft Agents

The airborne vehicle agents are so far only implemented on a low-fidelity model focusing solely on the cruise segment/operation of the vehicle. The parameters of the aircraft agent, including both flight and detection capabilities are:

- Cruise speed.
- Horizon visual sensor distance and angle.

- Infrared (IR) sensor distance and angle.
- Particle sensor distance and angle.

The aircraft agents variables are:

- Turn: Boolean variable (true or false) used in the flight path rules.

The current aircraft agents have a very simple set of rules to define the flight depending on it being a perimeter or non-perimeter agent. Perimeter agent rules are:

- If the actual coordinates are at the map X and/or Y limits, turn 90 degrees clockwise.

Non-perimeter agent rules are:

- If the actual coordinates are at the map X and/or Y limits, turn 180 degrees clockwise.

Figure 3 illustrates these rules and how they change in the current model if it is desired to adapt them to the wind angle β .

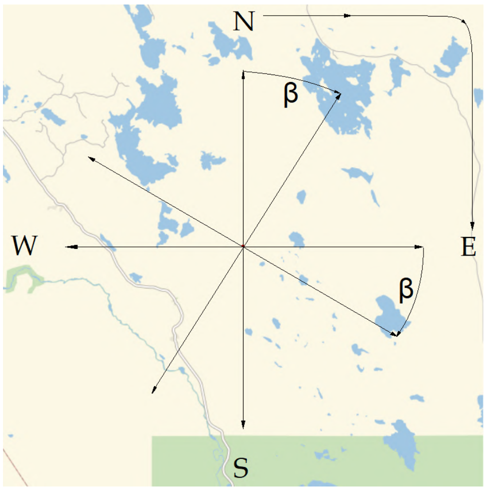


Figure 3 – Current flight trajectories allowed to the aircraft agents.

3.4 Topology

The geographic data necessary for the model can be obtained from any of the geographic information system (*GIS*) databases available nowadays, being many of them open source. The different geographic features are stored in raster data sets that discretise an area into a grid with values of different kind stored in each cell. Some common tools oriented to geographic data are, for example, ArcGIS, QGIS or OpenStreetMaps. The data is loaded to the model from a file in ASCII grid format [26]. The local slope angle, γ_S, of each patch is computed from the topographic slope angle obtained from the *GIS* data [23]:

$$tan\gamma_S = \|\nabla h\| = \sqrt{(\frac{\partial h}{\partial x})^2 + (\frac{\partial h}{\partial y})^2} \tag{14}$$

The topographic aspect γ_a, described in [23] as well is defined taken its sign into consideration as:

$$tan\gamma_a = \frac{-(\partial z/\partial y)}{(\partial z/\partial x)} \tag{15}$$

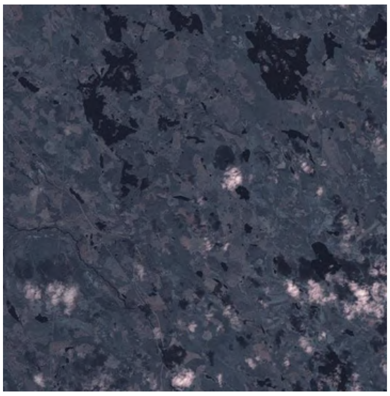
For the computation of the arctangent of a specific angle in the simulation model, the power series from [13] truncated at the 6th term are used:

$$arctan(x) = \begin{cases} x - x^3/3 + x^5/5 - x^7/7... & : -1 \leq x \leq 1 \\ \pi/2 - 1/x + 1/3x^3 - 1/5x^5... & : x > 1 \end{cases} \tag{16}$$

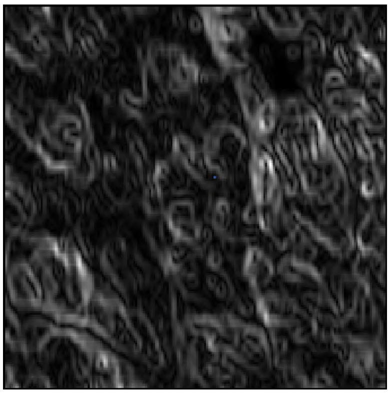
Notice that only the absolute value of the slope is needed since the topographic aspect is used to obtain whether the change is positive or negative.

4. Case Study Example

For the development of the Firefighting *SoS* framework an area of Sweden in the region of Gävleborg (Lon: 60.7806, Lat: 16.6553) has been studied. The area chosen for the *ABM* is approximately 19.4 km by 19.4 km and shown in Figure 4a, with the elevations for this area obtained with the help of [12] from *GIS* data. An example of the resulting local slopes (γ_S) from the data points imported into *NetLogo* is shown in Figure 4b. Table 8 shows the parameters used for the simulation setup.



(a) Satellite image of an area in Gävleborg (Sweden).



(b) Raster image of the local slopes obtained from geographical GIS data.

Figure 4 – Satellite photo of Gävleborg area in Sweden and the raster image of it.

The simulation runs were performed by starting the fires at points with equal distance between them,

Table 8 – Environmental parameters setup

Parameter	**Value**	**Comments**	**Units**
Number of trees	28400	Randomised positions	-
Temperature	10	-	[Celsius]
Wind angle	45	SW to NE	[degrees]
Wind velocity	11	-	[km/h]
Relative humidity	20	-	-
Day/night-time	Night	Visual sensor disabled	-
UAVs velocities	[150,90,90]	[UAV1,UAV2, UAV3]	[km/h]
Sensor distances	[1,1,1]	[Visual, particles, IR]	-
Sensor angles	[90,180,360]	[Visual, particles, IR]	-

resulting in a total of 100 simulations. The distances between each different simulated forest fire were 200 m (twice the sensor detection distance).
For comparison, two different *SoS* setups were tested with each comprising three UAVs:

- **Setup I: Static Flight Paths**

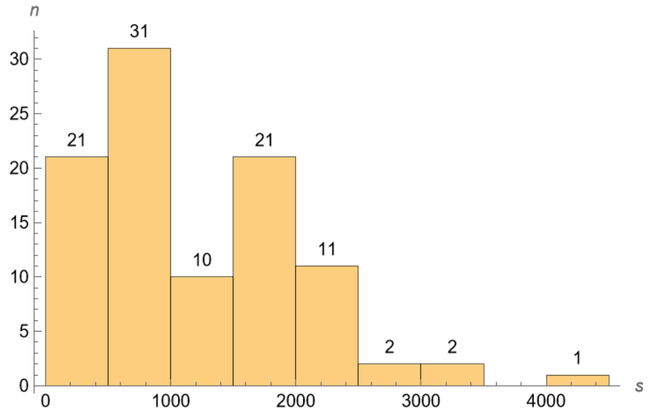
  Static flight paths with one UAV following the map edges clockwise, one UAV flying back and forth from North to South and one UAV flying back and forth from West to East.

- **Setup II: Wind-corrected Flight Paths**

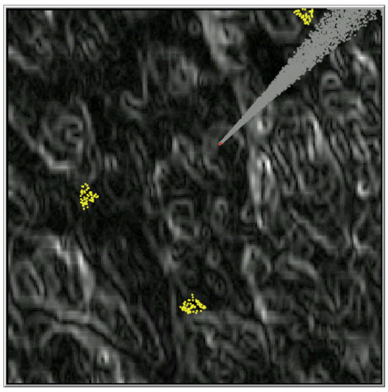
  Same setup as before but with the UAVs flying North-South and West-East shifting their angle accordingly to the wind angle, thus flying respectively parallel and perpendicular to it.

4.1 Static Flight Paths (Setup I) Results

Figure 5a shows a histogram for all simulated fire detection times, with an average detection time of 20 minutes. Figure 5b shows the smoke plume and the three detection cones for the UAVs for the case with the longest detection time (Setup I) of 67 minutes. Out of the 100 simulated cases, shown in figure 6 as a contour plot, 24 cases resulted in detection times longer than 30 minutes. These points are listed in Table 9 and depicted in figure 7b as checkpoints.



(a) Detection times histogram.



(b) Longest detection time scenario.

Figure 5 – Fixed flight paths detection times histogram and the scenario with the longest time needed for detection (Setup I).

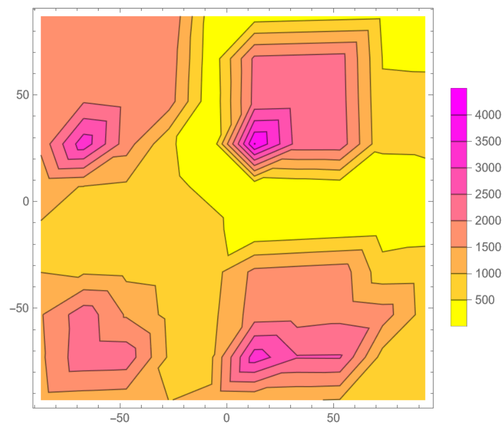
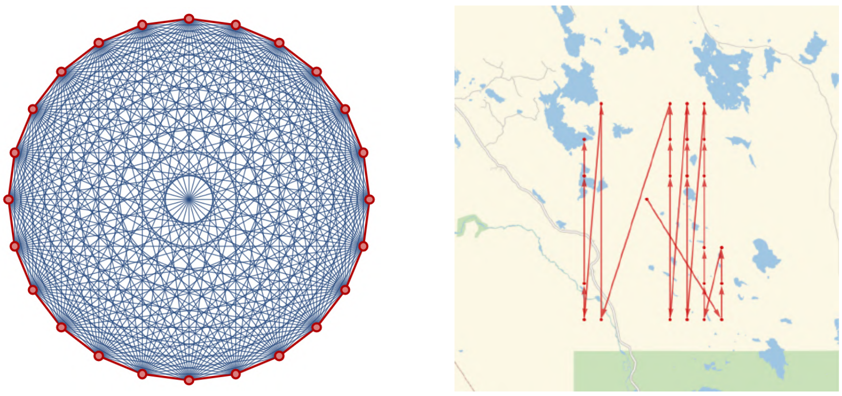


Figure 6 – Contour plot of needed time for detection depending on the fire initial point (Setup I).

Figure 7 shows a graph (Figure 7a) generated from the points in Table 9 with all the possible pairwise connections between them. The red and thicker path in it represents a computed Hamiltonian cycle in the graph (out of many other possible ones), which is used to compute a flight path for visiting only once all the aforementioned points. This flight path is shown in Figure 7b. The approximated length of the path is **122.08 km** (the distance flown by the detecting airplane in the worst-case scenario was 167.63 km). Figure 8 shows an alternative flight path (Figure 8b), with length **46.98 km**, which can be used for generating its correspondent Hamiltonian cycle shown in Figure 8a).

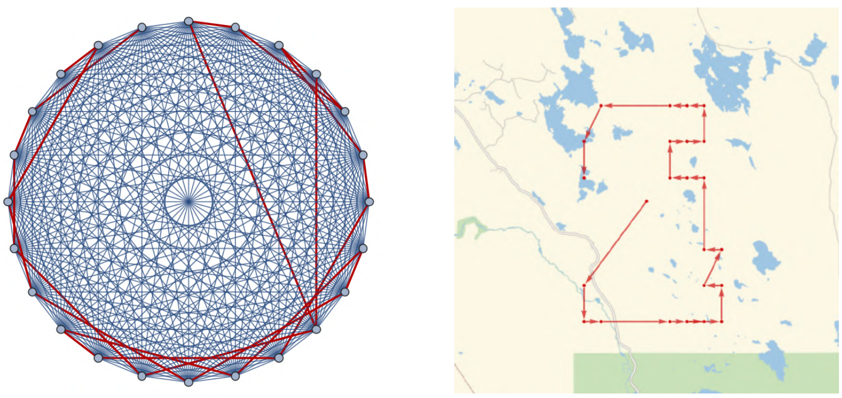
Table 9 – Points in the grid with a time needed for detection longer than 30 min (Setup I).

X - Coord.	Y - Coord.	Time [s]	X - Coord.	Y - Coord.	Time [s]
-67	87	1881	53	47	2248
-47	87	1931	-67	27	3309
-27	87	1977	13	27	4023
-67	67	1923	33	27	2309
-47	67	1883	53	27	2303
-27	67	1819	-67	-53	2294
13	67	2114	53	-53	1858
33	67	2162	-67	-73	2343
53	67	2211	-47	-73	2315
-67	47	1959	13	-73	3279
13	47	2150	33	-73	2532
33	47	2200	53	-73	2540



(a) Graph with a Hamiltonian cycle. (b) Computed flight path from 7a.

Figure 7 – Graph and Hamiltonian cycle built from the fire points with +30min needed for detection and the resultant flight path (7b) to visit them (Setup I).



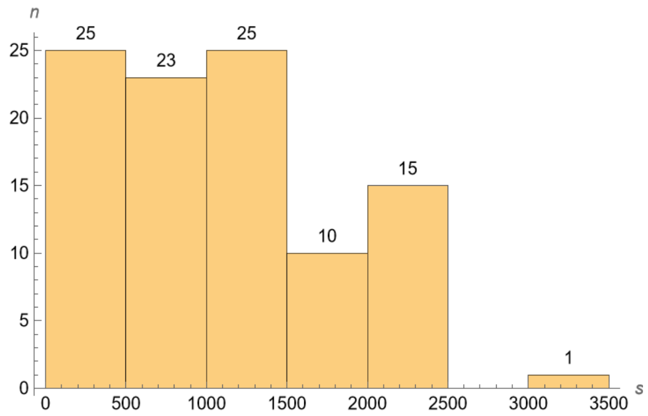
(a) Computed H. cycle from 8b. (b) Predefined flight path.

Figure 8 – Hamiltonian cycle (8a) computed from the checkpoints (8b) of a flight path to visit them (Setup I).

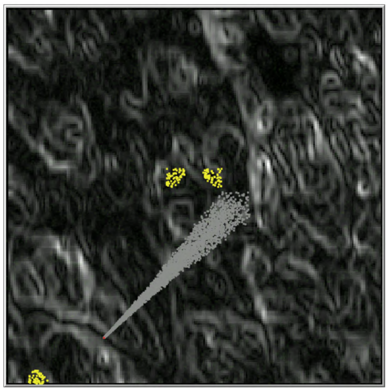
4.2 Wind-Corrected Flight Paths (Setup II) Results

Figure 9a shows a histogram for all simulated fire detection times with an average detection time of 18 minutes. Figure 9b shows the smoke plume, the three detection cones for the UAVs for the case with the longest detection time (Setup II) of 53 minutes. Out of the 100 simulated cases, shown in Figure 10 as a contour plot, 22 cases resulted in detection times longer than 30 minutes. These points are listed in Table 10 and depicted in figure 11b as checkpoints.

Figure 11 shows the graph (Figure 11a) generated from the points in Table 10 with all the possible pairwise connections between them. Again, the red and thicker path in it represents a computed Hamiltonian cycle in the graph (out of many other possible ones), which is used to compute a flight path for visiting only once all the aforementioned points. This flight path is shown in Figure 11b. The approximated length of the path is **120.06 km** (the distance flown by the detecting airplane in the worst-case scenario was 79 km). Figure 12 shows an alternative flight path (Figure 12b), with length **37.81 km**, which can be used for generating its correspondent Hamiltonian cycle shown in Figure 12a).



(a) Detection times histogram.



(b) Longest detection time scenario.

Figure 9 – Wind-corrected flight paths detection times histogram and scenario with longest time needed for detection (Setup II).

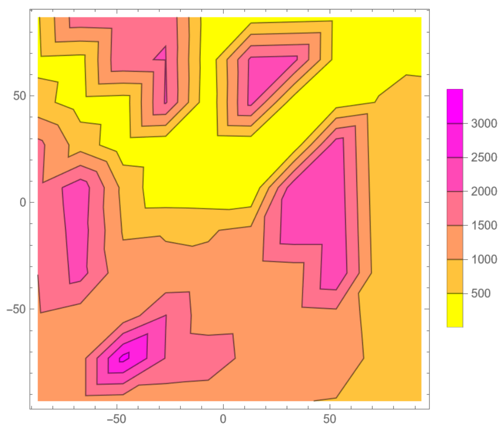
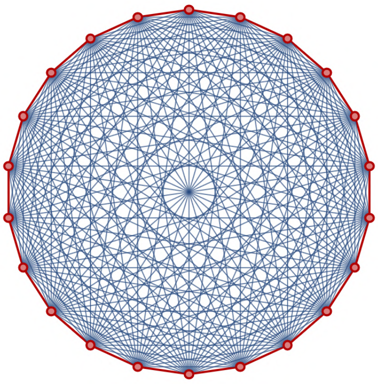


Figure 10 – Contour plot of needed time for detection depending on the fire initial point (Setup II).

Table 10 – Points in the grid with a time needed for detection longer than 30 min (Setup II).

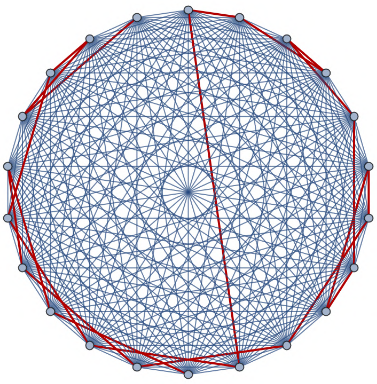
X - Coord.	Y - Coord.	Time [s]	X - Coord.	Y - Coord.	Time [s]
-67	87	1876	33	7	2346
-47	87	1924	53	7	2341
-27	87	1972	-67	-13	2238
-47	67	1968	33	-13	2390
-27	67	2013	53	-13	2389
13	67	2111	-67	-33	2184
33	67	2157	53	-33	2440
-27	47	2052	-27	-53	2012
13	47	2149	-47	-73	3159
53	27	2295	-27	-73	1979
-67	7	2303	-7	-73	1902



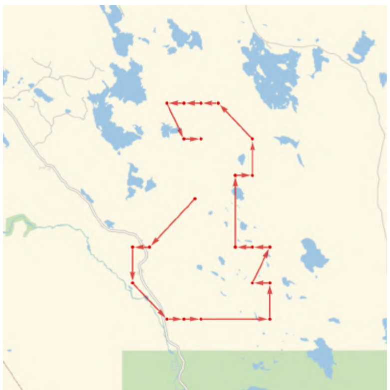
(a) Graph with Hamiltonian cycle.

(b) Resultant flight path from 11a.

Figure 11 – Graph and Hamiltonian cycle built from the fire points with +30min needed for detection and the resultant flight path (11b) to visit them (Setup II).



(a) Computed H. cycle from 12b.



(b) Predefined flight path.

Figure 12 – Hamiltonian cycle (12a) computed from the checkpoints (12b) of a flight path to visit them (Setup II).

5. Discussion

The model is considered suitable for wildfire simulations after comparison with data and figures in [1] for validation. Figure 13 shows the evolution of a reported fire in 2014, with an approximated time lapse of 210 minutes. The data included measured values of wind velocity, temperature and relative

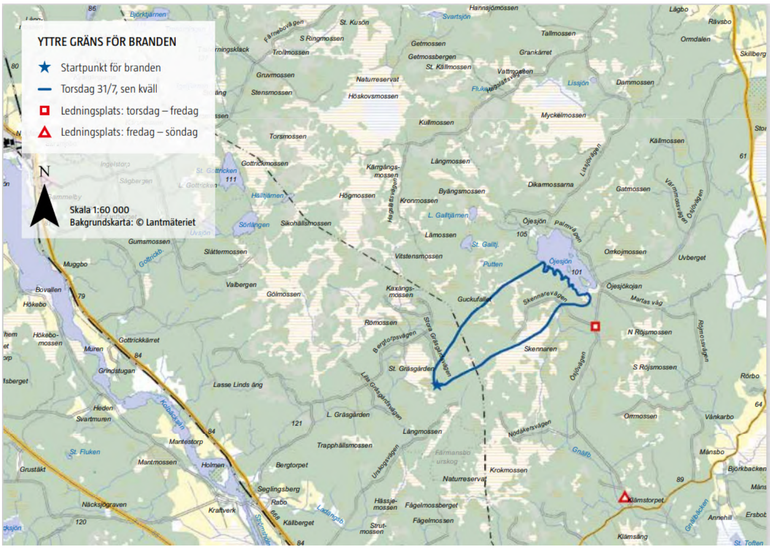


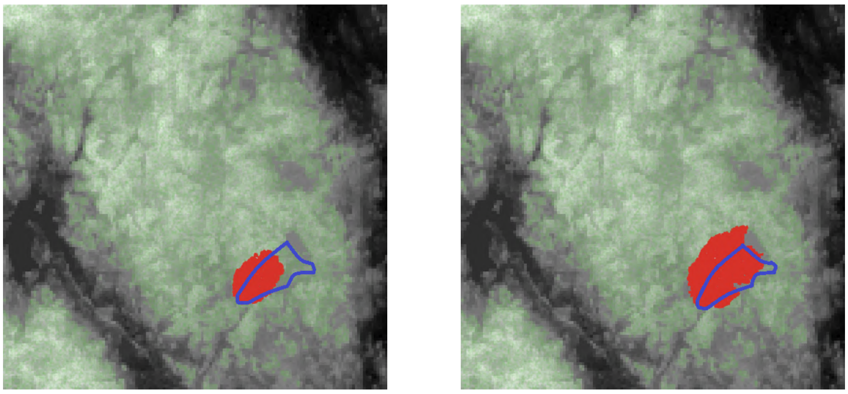
Figure 13 – Figure from [1] of the forest fire in Västmanland (Sweden) in 2014.

humidity. These values were not available in the same time scale as the simulation, thus the comparison was done first by using an average of the data available, with the resultant area shown in Figure 14a, and second by following approximately the variations in time of the data, with the resultant area shown in Figure 14b.

Both cases show results close the reference area in Figure 13. The case of the simulation with constant values for the environment shows a smaller area than the reference, consistent with a resultant spread rate less favorable for the fire. On the other hand, the case of variable environment shows a slightly over-predicted area, still within a range assumed acceptable for a simulation run with just approximated values of the available data.

The case of the smoke plume is more challenging to validate. An example of smoke plume is shown in Figure 15. Whilst an accurate comparison is very difficult to achieve and no specific comparison has been done for it, the qualitative similarity with Figure 5b in terms of shape is considered acceptable for a simplified model of smoke plumes. Hence, the *ABM* at hand is considered valid for the study of *SoSs*

The detection tactics in the studied case have been shown to be relevant for definition of capabilities and additional requirements. The wind-corrected *SoS* shows a reduction in the number of fires that will need more than 30 minutes for being detected from 24, down to 22, but more importantly, it shows a reduction of the average detection time, from 20 minutes in the Setup I, down to 18 minutes in the Setup II. Moreover, Figures 5a and 9a show that there is also a reduction in the longest detection time, from 67 minutes in the Setup I, down to 53 minutes in the Setup II. The detection at this point is assumed to be always true within the sensor range, one of the simplifications of the model that could be enhanced in the future, either by artificial intelligence to identify a possible fire or by human judgement to be included in the loop. The results shown in Figures 7 and 11 show how graph theory [11] could be applied in the design of *SoSs* and the most suitable combination of constituent systems. It becomes then an optimization problem for finding the shortest paths (such as Figures 8 and 12), where the reductions have been from *122.08 km* to *46.98 km*, and from *120.06 km* to *37.81 km* respectively. This raises, however, the question of the most desirable situation: an additional system in the *SoS* capable of following the longer path and inspecting more area during its way to the desired points or, on the other hand, a system to scrutinise the desired points without being concerned of the areas in between in the least amount of time possible? In either case, the results provide additional information for comparison of capabilities among a collection of *SoS* and



(a) Simulation with constant environment for the time lapse using averaged data.

(b) Simulation with approximated environmental evolution from available data.

Figure 14 – Wildfire simulations using the data available in [1].

Figure 15 – Satellite photo of a fire plume in 2021 from [20]

for the definition of the requirements of the constituent systems.

6. Conclusion and Future Work

The fidelity of the present *ABM* can be enhanced with the inclusion of more *GIS* data to the model, i.e., mapping of forests, or water areas, such as rivers or lakes, roads, train railways, etc. For a holistic view, a more complete *ABM* and a more detailed framework is needed. In terms of future research, the question to answer is what level of detail is needed for each matter: a lower level of detail for running a large number of scenarios and their *SoS*; whilst a level as high as possible for a specific evaluation would be preferred. Being able to modify the *ABM* in such manner would be relevant as well. The study of *SoS* is at early stages still, but identifying common needs for performing it, independently of the *SoS* background, is important for achieving some level of standardisation.

The constituent systems composing the *SoS* in the current *ABM* allow different levels of fidelity. For example, it can be assumed that their control system is capable of sustained level flight and constant velocity. However, a certain level of detail is needed for everything that affects the fuel or energy consumption, such as drag or energy requirements and weight of the subsystems included. Therefore, the study of capabilities and evaluation of needs fulfilment in terms of the design space for a *SoS* could be performed in the current *ABM* and its future evolution. Likewise, subsystems in the constituent systems may need a higher level of detail than the actual system including them. The impact of their resolution, position inside the system, energy consumption, weight or reliability on the *SoS* is also a research topic by itself. However, the need to keep track of the different comparisons and changes done for achieving traceability opens the inclusion of the use of, for instance, ontology-based tools [15].

6.1 Data analysis and visualization

Beyond the simulation animation, important for observing the evolution of the *SoS* in time, the data generated by the simulation needs to be evaluated for discarding irrelevant values first, and analyzing of the potentially relevant afterwards. The concept of *visual analytics*, being it a field of study on its own as well, is useful for preliminary evaluation. Values such as time needed to reach a goal and the monetary resources used in the process are straightforward, but other options should be considered and weighted. An *ABS* representing an *SoS* can show behavioural patterns very difficult to predict and therefore enable the study of secondary effects to support decision making, data-based modelling, increased value comparison, etc., but also for an attempt to achieve traceability and standardization of *SoS* frameworks.

6.2 Design and optimization

The current study of *SoS* framework with the present *ABM* allows also design space evaluations and different kinds of optimization studies. They are, however, two concepts that are very related. For example, if the fire risk in a certain area is given by a probability density function obtained from *ABS*, it is possible to redefine different priorities for visiting points in a map based on factors such as proximity to urban areas or tree density.

Graph theory has been used in the study of *SoS* here for comparing different paths. By applying this method to define flight paths, it is possible to find routes of minimal length similar to *Dubins paths* [9] and therefore range optimization of the constituent systems.

For *SoS* have different levels of control and inherent behavioural characteristics, research also suggests [3] that the study of game theory and cooperation could be beneficial for strategy optimization within the *SoS* framework as well. It is a concept not only interesting in relation to ABM, but also for AI development that can be used for the design of unmanned aerial vehicles *UAVs*.

7. Acknowledgments

The authors would like to acknowledge the Swedish Innovation agency (VINNOVA) for its financial support through the grant 2019-05371

8. Contact Author Email Address

To contact the authors send an email to: jorge.lovaco@liu.se

9. Copyright Statement

References

References

[1] Myndigheten för samhällsskydd och beredskap (MSB). *Skogsbranden i Västmanland 2014*. MSB798, 2015. ISBN: 978-91-7383-527-5.

[2] Gary L Achtemeier et al. "Modeling smoke plume-rise and dispersion from southern United States prescribed burns with Daysmoke". In: *Atmosphere* 2.3 (2011), pp. 358–388.

[3] Sutee Anantsuksomsri and Nij Tontisirin. "A spatial agent-based model of a congestion game: evolutionary game theory in space". In: *The Annals of Regional Science* 57.2 (2016), pp. 371–391.

[4] Hal E Anderson. *Predicting wind-driven wind land fire size and shape*. Vol. 305. US Department of Agriculture, Forest Service, Intermountain Forest and Range ..., 1983.

[5] Panagiotis Barmpoutis et al. "A review on early forest fire detection systems using optical remote sensing". In: *Sensors* 20.22 (2020), p. 6442.

[6] Miguel G Cruz and Martin E Alexander. “The 10% wind speed rule of thumb for estimating a wildfire’s forward rate of spread in forests and shrublands”. In: *Annals of Forest Science* 76.2 (2019), pp. 1–11.

[7] Scott De Marchi and Scott E Page. “Agent-based models”. In: *Annual Review of political science* 17 (2014), pp. 1–20.

[8] Christian Derksen, Cherif Branki, and Rainer Unland. “A framework for agent-based simulations of hybrid energy infrastructures”. In: *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE. 2012, pp. 1293–1299.

[9] Lester E Dubins. “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents”. In: *American Journal of mathematics* 79.3 (1957), pp. 497–516.

[10] Wallace L Fons. “Analysis of fire spread in light forest fuels”. In: *Journal of Agricultural Research* 72.3 (1946), pp. 93–121.

[11] Willie K Harrison. “The role of graph theory in system of systems engineering”. In: *IEEE Access* 4 (2016), pp. 1716–1742.

[12] Wolfram Research Inc. *Mathematica, Version 12.1*. Champaign, IL, 2020.

[13] Alan Jeffrey and Hui Hui Dai. *Handbook of mathematical formulas and integrals*. Elsevier, 2008.

[14] George M Jemison. “Some Principles of Visibility and their Application to Forest Fire Detection”. In: *US Dept. Agr. Tech. Bul* 954 (1948), p. 61.

[15] Ludvig Knöös Franzén. “An Ontological and Reasoning Approach to System of Systems”. PhD thesis. Linköping University Electronic Press, 2021.

[16] Ludvig Knöös Franzén et al. “A System of Systems Approach for Search and Rescue Missions”. In: *AIAA Scitech 2020 Forum*. 2020, p. 0455.

[17] Petter Krus. “Aircraft System Optimization and Analysis for Traceability in Design”. In: *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2006, p. 7017.

[18] Dominique Luzeaux and Jean-Rene Ruault. *Systems of systems*. John Wiley & Sons, 2013.

[19] Kilian J Murphy, Simone Ciuti, and Adam Kane. “An introduction to agent-based models as an accessible surrogate to field-based research and teaching”. In: *Ecology and evolution* 10.22 (2020), pp. 12482–12498.

[20] NASA - Earth Observatory. *Alisal Fire Rages Near Santa Barbara*. 2021. URL: `https://earthobservatory.nasa.gov/images/148950/alisal-fire-rages-near-santa-barbara` (visited on 05/31/2022).

[21] José CF Pereira et al. “Calculation of spotting particles maximum distance in idealised forest fire scenarios”. In: *Journal of Combustion* 2015 (2015).

[22] James Edmund Reeb et al. “Wood and moisture relationships”. In: (1995).

[23] Jason J Sharples. “Review of formal methodologies for wind–slope correction of wildfire rate of spread”. In: *International Journal of Wildland Fire* 17.2 (2008), pp. 179–193.

[24] Michael A Storey et al. “Drivers of long-distance spotting during wildfires in south-eastern Australia”. In: *International journal of wildland fire* 29.6 (2020), pp. 459–472.

[25] Seth Tisue and Uri Wilensky. “Netlogo: A simple environment for modeling complexity”. In: *International conference on complex systems*. Vol. 21. Boston, MA. 2004, pp. 16–21.

[26] Wikipedia. *Esri grid.* 2022. URL: `https://en.wikipedia.org/wiki/Esri_grid` (visited on 05/31/2022).

[27] U Wilensky. *NetLogo*. 1999.

[28] U Wilensky. *NetLogo Wolf Sheep Predation model.* 1997.