Self-Organising Swarms of Firefighting Drones: Harnessing the Power of Collective Intelligence in Decentralised Multi-Robot Systems

Mauro S. Innocente[a], Paolo Grasso[a]

[a]*Autonomous Vehicles & Artificial Intelligence Laboratory (AVAILab), Research Institute for Future Transport and Cities, Coventry University, Coventry, UK.*

Abstract

Swarm Intelligence (SI) is concerned with the collective behaviour that emerges from decentralised self-organising systems, whilst Swarm Robotics (SR) is an approach to the self-coordination of large numbers of simple robots which emerged as the application of SI to multi-robot systems. Given the increasing severity and frequency of occurrence of wildfires and the hazardous nature of fighting their propagation, the use of disposable inexpensive robots in place of humans is of special interest. This paper demonstrates the feasibility and potential of employing SR to fight fires autonomously, with a focus on the self-coordination mechanisms for the desired firefighting behaviour to emerge. Thus, an efficient physics-based model of fire propagation and a self-organisation algorithm for swarms of firefighting drones are developed and coupled, with the collaborative behaviour based on a particle swarm algorithm adapted to individuals operating within physical dynamic environments of high severity and frequency of change. Numerical experiments demonstrate that the proposed self-organising system is effective, scalable and fault-tolerant, comprising a promising approach to dealing with the suppression of wildfires – one of the world's most pressing challenges of our time.

Keywords: Self-organisation, Particle swarm, Fire spread modelling, Swarm robotics, Autonomous unmanned aerial vehicles

1. Introduction

A wildfire is an unplanned and uncontrolled vegetation fire which can have devastating health, social, economic and environmental impacts. The deadliest wildfires in Portugal's history erupted in 2017, killing 66 and injuring 204 people. In 2018, several major series of wildfires broke out around the world, including in the United States (US), Canada, Australia, Greece, Portugal and the United kingdom (UK): California saw the largest wildfires on record; British Columbia saw the largest total burn-area during wildfire season on record; Sydney's bushfire season started two months early, in winter; more than 80 people were killed in Athens; forest fires wreaked havoc in the Algarve region; whilst a record-breaking series of wildfires burnt across the UK. In February 2019, New Zealand saw the worst wildfires in over 50 years, with 155 firefighters, 23 helicopters and three planes deployed to tackle the blaze. One helicopter crashed fighting the Nelson wildfire, with the pilot suffering moderate injuries. The Forestry Commission predicts that destructive wildfires will increase in frequency due to increased land-use pressure and climate change [24]. The development of more effective and safer means to fight wildfires is one of the world's most pressing challenges of our time.

Unmanned Aerial Vehicle (UAV) technology has progressed rapidly for the past two decades, extending its capabilities and the kinds of problems it can help tackle. Modern UAVs –a.k.a. drones– can be equipped with a range of advanced cameras and sensors which enable them to operate in remote areas, dangerous environments, and even through solid smoke. Current applications include aerial photography and filming, information gathering for human decision-makers, provision of essential supplies, support for search and rescue operations, mapping of inaccessible locations, field surveying, and crop health monitoring. With regards to firefighting operations, drone technology has been applied to forest surveillance, building fire risk maps, forest fire detection and monitoring [15], post-fire recovery monitoring [84], bushfire hotspot detection [30], and support for disaster relief operations. Compared with their use in forest fire monitoring and detection, research and development on UAV-based fire suppression is still scarce [89]. Yet, given the hazardous nature of the activity, fighting fires using UAVs in place of humans is of special interest.

Swarm Intelligence (SI) is a route to Artificial Intelligence (AI) which stems from decentralised and self-organising behaviour observed in groups of social animals in nature. By way of collaboration, a form of collective intelligence emerges enabling them to accomplish tasks that are far beyond the aggregation of their individual capabilities. SI is the branch of AI that deals with the collective behaviour that emerges from decentralised self-organising systems, where individuals only interact locally with one another and with the environment. Swarm Robotics (SR) is an approach to the self-coordination of large numbers of simple robots which emerged as the application of SI to multi-robot systems. It differs from other SI studies in that it emphasises the physical embodiment of individuals [74], and from distributed robotics in that it promotes scalability.

There is a limited number of UAVs which can be remotely controlled and coordinated to operate simultaneously, therefore restricting the achievable fire suppression capabilities. There

Email address: Mauro.S.Innocente@gmail.com (Mauro S. Innocente)

Preprint submitted to Journal of Computational Science, Special Issue: Soft Computing for Swarm Robotics

Accepted version of the manuscript.
Published version: Journal of Computational Science 34 (2019) 80–101, doi: 10.1016/j.jocs.2019.04.009

are also difficulties associated with centralised communication with ground control during wildfire events. Conversely, swarm robotic systems allow for a high number of self-coordinating agents with only local drone-to-drone communication and no central control. Furthermore, the use of swarms of decentralised collaborative and self-organising robots results in a robust and resilient system with collective decision-making able to cope with uncertainty, errors, local perturbations, and the failure or loss of a few units. While the use of drones to support firefighting operations is fast becoming common practice, the design of self-organising swarms of drones to directly engage in the suppression of fires remains notably unexplored.

This paper aims to demonstrate the feasibility and potential of employing SR to fight wildfires autonomously. The focus is not on the design of the physical robots but on their self-coordination mechanisms for the desired firefighting behaviour to emerge. To this end, an efficient yet realistic physics-based model of wildfire propagation and a self-organisation algorithm for swarms of firefighting drones are developed and coupled, with the collaborative behaviour based on a particle swarm algorithm adapted to individuals operating in physical dynamic environments of high severity and frequency of change.

Section 2 of this paper presents background and related work on fire propagation models, SI, SR, and the particle swarm algorithm. Section 3 presents the proposed efficient physics-based model of wildfire propagation, whilst section 4 describes the proposed system of self-organising swarms of firefighting drones. Numerical experiments are carried out in section 5 to calibrate the fire propagation model in terms of the area affected against predictions obtained by a commercial simulator, to observe unchecked fire propagation, to test the proposed fire suppression system in terms of expected scalability and fault tolerance, and to study the effect of the inertia weight as a control coefficient. Conclusions and future work are discussed in section 6. All figures in this article are available in [dataset] [36].

2. Background and Related Work

2.1. Fire Propagation Models

Modelling fires and their propagation is a remarkably challenging task due to the complex combination of interdependent physical phenomena involved that take place at different scales. Depending on the features considered and techniques used, different taxonomies of mathematical models can be devised.

Sullivan [80, 81, 82] proposes three broad categories, which in fact become six: 1) *physical models*, which attempt to represent the physics and chemistry of fire spread; 2) *quasi-physical models*, which attempt to represent only the physics; 3) *empirical models*, which contain no physical basis; 4) *quasi-empirical model*, which use some physical framework upon which to base the empirical model; 5) *simulation models*, which implement the preceding types of models in a computer simulation environment; and 6) *mathematical analogous models*, which use mathematical precepts rather than physical ones.

In this paper, three classifications of mathematical models are proposed in Fig. 1, which are somewhat general yet especially suitable for fire propagation models.

Perhaps the most obvious differentiation is between *mechanistic* and **Data-Driven Models**. While the latter are built so as to fit data, mechanistic models are developed based on a set of *governing laws* or mechanisms. If these laws are directly associated with the phenomena or system being modelled, they are referred to as **Theoretical Models** –which are often *physics-based*. Otherwise, they are referred to as **Mechanistic Surrogate Models**, since the governing laws are seemingly unrelated to the problem in question yet predictions appear sufficiently accurate. It seems appropriate to quote Box et al. [12] here: "All models are wrong, but some are useful".

Theoretical models derived purely from *first principles* are said to be *white-box* models whilst data-driven models which are purely *data-fitting* –completely independent of theories– are said to be *black-box* models (*phenomenological*, *empirical*). The combination of governing laws and empirical data leads to *grey-box* models (*semi-empirical*). Typically, the latter are viewed as theoretical if mostly based on governing laws with some parameters calibrated using empirical data (*light-grey-box*) and as data-driven if mostly driven by data but incorporating some principles and laws associated with theories (*dark-grey-box*, *model-fitting*). If a data-driven model is built to fit data generated using another model (typically theoretical and high-fidelity), the result is a *data-driven surrogate model*, or simply *surrogate model*. Mechanistic surrogate models are those designed disregarding the theories underlying the system or phenomena being modelled, yet based on a few leading assumptions which may be somewhat physics-based.

2.1.1. Data-Driven Models

Data-driven models are based on observation and experimentation, either constructing a model directly from the data (empirical, black-box, data-fitting) or fitting a pre-defined model to the data (semi-empirical, dark-grey-box, model-fitting). They may be used to simply model available data or as the basis for theoretical developments and their subsequent testing.

In the early models of wildfire propagation, the focus was on the determination of key characteristics identified to describe the behaviour of wildfires such as the local rate of spread (RoS) at the headfire, the height and angle of the flames, or the rate of increase of the affected area. Hence they basically consist of one-dimensional (1D) models of fire behaviour. A prominent example is Rothermel's model [70, 5], which predicts the RoS of the headfire in the direction of the wind in an environment specified by fuel, weather and topography descriptors [64]. It is based on a heat balance model using data obtained from wind tunnel experiments in artificial fuel beds of varying characteristics and from field experiments of grassfires under varying wind conditions [5, 81]. Thus, Rothermel's model is a *dark-grey-box model* in Fig. 1 (data-driven, model-fitting), whilst it is classified as *semi-physical* in [64] and as *quasi-empirical* in [81].

In the US, Rothermel's model forms the basis of the National Fire Danger Rating System [5, 81] and has been incorporated into the fire modelling system BehavePlus [4] and the Fire Area Simulator FARSITE [27]. In Australia, Cheney et al.'s empirical model [18] is used in the CSIRO Grassland Fire Spread Meter to predict fire behaviour in grasslands [81]. In Canada, the

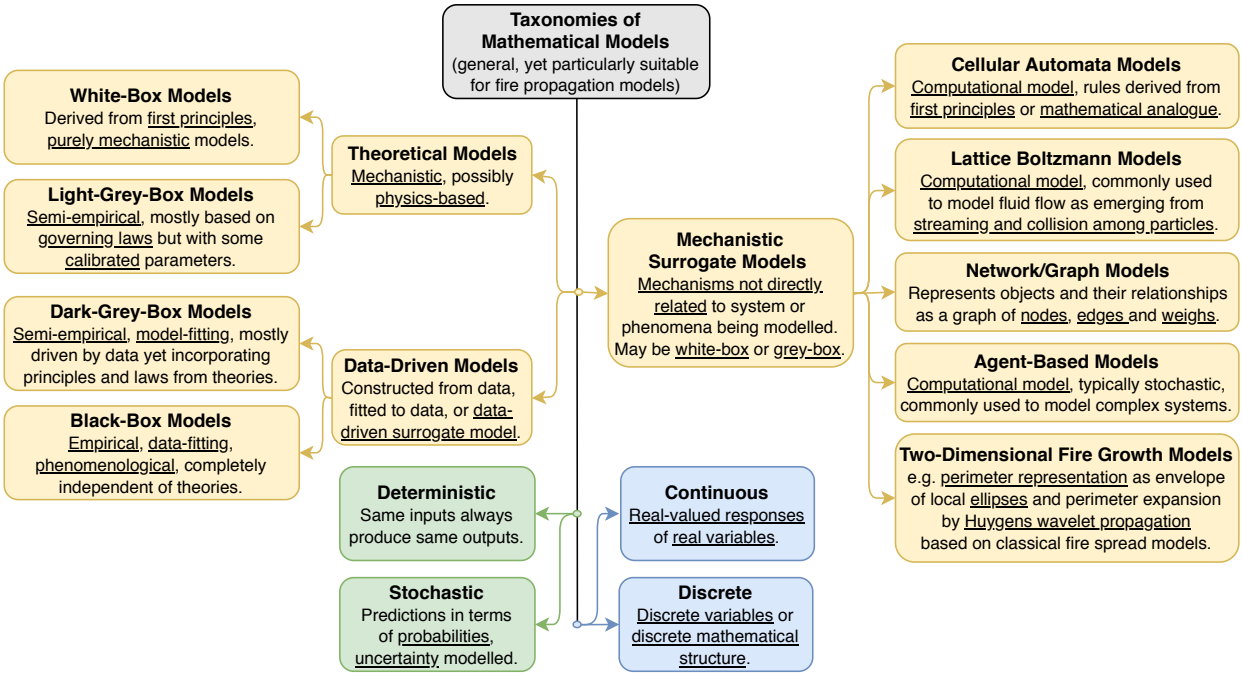


Figure 1: Proposed taxonomies of mathematical models, particularly suitable for fire propagation models. A given model may belong to more than one class; e.g. a cellular automata model is based on difference equations (discrete), may be a mechanistic surrogate model, derived from first principles (white-box) or data-driven (model-fitting), and may be either deterministic or stochastic (typically the latter).

quasi-empirical Fire Behaviour Prediction system [32] forms part of the Canadian Forest Fire Danger Rating System. For a comprehensive review of data-driven models, refer to [81].

While useful under specific conditions, data-driven models' predictions cannot easily be extrapolated beyond the conditions under which they were developed [64].

2.1.2. Theoretical Models

When describing physical phenomena, theoretical models typically present mathematical formulations in terms of coupled partial differential equations (PDEs). Physics-based white-box models are often 3D, developed from first principles, and include balances of mass, momentum and energy. In the context of fire propagation modelling, they also account for chemical reactions involved in combustion processes, and therefore involve convection-reaction-diffusion-radiation equations with model parameters aimed to be mathematically derived. Prominent examples are the diphasic multiscale model in [75], FIRETEC [52], FDS [56] and WFDS [58]. Similarly, physics-based light-grey-box models tend to be 2D, based on simple balance laws such as energy balance (e.g. [55, 26, 31]), whilst also requiring empirical parameters and experimental data to calibrate them for particular conditions.

For a model to be useful, it must be validated. Even advanced models developed from first principles make a number of assumptions and simplifications such as averaging small-scale processes, disregarding radiant heat from flames, or modelling only gas-phase combustion. Others apply reduction techniques to make the models numerically tractable and operationally feasible (e.g. [55, 2]). Therefore, even multiphysics, multiphase and multiscale theoretical models cannot be exclusively bottom up from first principles, and require some degree of calibration to be validated so as to produce reliable predictions. For example, notice the calibrations during validation of FDS in [57]. Nonetheless, calibrated parameters in a physics-based model should still have physical meaning and their values should remain within realistic intervals. For a comprehensive review of this type of models, refer to [58, 80].

2.1.3. Mechanistic Surrogate Models

Mechanistic surrogate models do not closely follow the laws of physics associated with the phenomena or system being modelled. Instead, a different set of governing laws are observed to predict the desired responses with sufficient precision. While this makes them computationally efficient, the drawback is that the use of the model is limited to making that specific prediction. For example, a mechanistic surrogate model may be able to predict the spatial evolution of a fire front without any regard to the physics involved, and therefore cannot be later used to predict temperatures, to interact with an atmosphere model, or to study the impact of a given fire suppressant. Examples related to fire spread modelling are *Cellular Automata Models*, *Lattice Boltzmann Models*, *Network Models*, *Agent-Based Models*, and *Two-Dimensional Fire Growth Models*.

A Cellular Automata (CA) model is a decentralised spatially extended system consisting of a large number of identical components of simple geometry with local connectivity. Two main components can be identified: 1) *the cellular space* and 2) *the transition rule*. The former is a lattice of a certain number of identical finite-state machines whereas the latter returns the new

state based on the current state and on those of the neighbouring cells. While a CA model may be physics-based, it usually comprises a mechanistic surrogate model. Since it is a dynamic system discrete in time and space, its computer implementation is straightforward. It also allows for the seamless introduction of stochasticity into the transition rule and of percolation threshold analysis. Examples are [23, 61, 1, 71, 25, 60].

A Lattice Boltzmann method is a promising approach for the simulation of fluid flow [79, 7] alternative to the conventional approach of numerically solving the Navier-Stokes equations. Lattice-Boltzmann models (LBMs) represent the fluid as a large but finite number of particles undergoing propagation and collision processes. The domain is represented by a lattice mesh at which nodes a reduced number of particles are confined. This model consists of a discretised representation of the Boltzmann transport equation that relates the particles' distribution to their velocities by means of a collision operator. Examples of combustion simulations using this approach are [87, 16, 19].

Some authors have attempted to model fire propagation using *small-world networks*, where a square lattice is used to model the short-range phenomena (e.g. radiation, convection, diffusion) whilst long-range connections are added to model fire-spotting processes such as firebrands transported downwind starting spot-fires in wildlands [65] or fire transmission through walls and ducts in fire-spread onboard naval vessels [45].

An agent-based model (ABM) is a class of computational models that simulate the local actions and interactions of autonomous agents to study the behaviour that emerges at the system level. ABMs describe the system or phenomena of interest using agents, an environment, and both agent-agent and agent-environment interactions. If applied to model fire propagation, agents may represent the fire front travelling across a raster-based simulated environment (e.g. [22]).

Specific to fire propagation modelling, there is a family of mechanistic surrogate models based on a two-dimensional representation of the fire perimeter followed by some law governing its expansion. Even if they include large data-driven components, they are classified as *mechanistic surrogate models* here because they incorporate crucial mechanisms which are not directly linked to fire dynamics. Their objective is to predict the extent of a fire rather than its quantitative behaviour [64], involving two distinct processes: 1) representing the fire front, and 2) propagating that perimeter. In this paper, we group these models under the denomination of *Two-Dimensional Fire Growth Models*. This class can be further subdivided into two subclasses: 1) *Level-set-based models* (LSM) and 2) *Marker-based models* (MMs) [11]. LSMs predict the advance of a fire front according to some governing law that specifies the RoS of an infinitesimal segment of fire perimeter arc normal to itself [54]. Instead, MMs typically estimate the fire RoS with Rothermel's model [70], assume a local elliptical model for the fire shape, and then follow Huygens' principle with elliptical expansion to propagate the perimeter (e.g. [68, 29]). Each marker along the perimeter behaves as a new ignition point for the next time step. The basic propagation geometry is a local ellipse with its rear focus centered on each marker, which size, shape and orientation depend on the fuel type, wind intensity and local slope. After one time increment, the new perimeter is given by the outer curve that envelopes all local ellipses. Prominent examples of MMs are the popular Fire Area Simulator (*FARSITE*) [27], the Canadian wildland fire growth simulator *Prometheus* [85], and the Australian Bushfire Risk Management Tool *Phoenix* [83]. Rio et al. [69] combines Rothermel's and Richard's [68] models into a forward model, and an inverse modelling approach that minimises the differences between the predictions of the forward model and sensor observations (i.e. a data-driven assisted mechanistic surrogate model).

2.1.4. Other Classifications

As shown in Fig. 1, other classification criteria are possible such as differentiating between *continuous* and *discrete* or between *deterministic* and *stochastic* models irrespective of whether it is theoretical, data-driven or a mechanistic surrogate. Continuous models represent the system or phenomena under study in a continuous manner, with real-valued functions (or responses) of real variables. Conversely, discrete models make use of discrete variables and/or discrete mathematical structures. In turn, deterministic models always return the same predictions under the same circumstances, as they are governed by equations that describe exactly how the system or phenomena behaves. In contrast, a stochastic model will return probabilistic predictions, producing different results depending on the values that random variables take in each realisation.

2.2. Swarm Intelligence

Swarm Intelligence (SI) is a paradigm of AI which stems from decentralised and self-organising behaviour observed in groups of simple social animals in nature such as ant, termite and bee colonies, fish schools and bird flocks. By way of collaboration, a form of collective intelligence emerges enabling these animals to accomplish tasks that are beyond the simple aggregation of their individual capabilities. That is, the whole is more than the sum of its parts. Thus, SI is the branch of AI that deals with the collective behaviour that emerges from decentralised self-organising systems. Self-Organisation occurs with no central control or sense of purpose, as individuals only interact locally with one another and with the environment inducing the emergence of coherent global patterns.

SI is the emergent collective intelligence of groups of simple agents [10]. As an AI discipline, it is concerned with the design of intelligent multi-agent systems. A SI system should satisfy the following conditions:

a) Be composed of a number of simple agents or individuals, as some critical mass is required for self-organisation to occur.
b) Be composed of agents or individuals who are similar to one another, typically identical or belonging to a few typologies.
c) Have local interactions based on simple behavioural rules that exploit information exchanged locally among individuals in a direct manner or indirectly through the environment. This indirect exchange is called *stigmergy*, where the environment acts as state memory. Interactions are typically probabilistic, introducing creativity into the system.

d) Exhibit an emergent global behaviour which results from the interactions of individuals with one another and with the environment. This self-organised intelligent behaviour at the swarm level is not known at the individual level.

These characteristics make SI systems scalable, parallel, robust and fault-tolerant. Scalability guarantees that the system can change size without redefining its behaviour. Since interactions are local, individual behaviour is marginally affected by changes in the swarm size. In turn, the population-based nature of these systems results in a parallel search that acquires information in a distributed manner. This makes them robust to local perturbations whilst the exchanges of information among agents enlarge the pool of knowledge decreasing uncertainty. Fault tolerance is due to their decentralised, self-organised and scalable nature as well as to the similarity among agents. If an agent is faulty or removed, the system does not cease to function: it can either continue to work under the reduced swarm-size, or another agent can be incorporated as a substitute.

Critical concepts underlying the field of *Swarm Intelligence* are *Emergence*, *Complexity*, *Self-Organisation* and *Stigmergy*. A detailed discussion on these topics is beyond the scope of this paper, and the reader is referred to the excellent articles compiled in [6] and to [10, 34, 35, 59]. It is simply mentioned here that self-organisation is a process through which coherent global patterns emerge from local interactions among lower-level components of the system. For this to occur, numerous interactions, stochasticity, and a combination of positive and negative feedbacks are essential. Positive feedback is generated through autocatalytic behaviours whereas negative feedback is a counterbalancing mechanism that typically stems from a depletion of resources. [10, 74]

It should be noted that there are several terms used indistinctly in this section and in the literature whose definitions are rather loose and frequently conflicting. For instance, the terms *cooperation* and *collaboration* do not have identical meaning. The same is true for *cooperative* and *collective* decision-making, *distributed* and *decentralised* systems, and *collective* and *swarm* intelligence. While different, it is argued that most of these concepts apply to SI systems nonetheless, and the issue is not discussed further in this article.

2.3. Swarm Robotics

Swarm Robotics (SR) is an approach to the self-coordination of large numbers of simple, relatively inexpensive robots which emerged as the application of SI to multi-robot systems. Different from other SI studies, emphasis is on the physical embodiment of individuals [74]. SR systems differ from distributed robotic ones in that the former are scalable, which means that performance can be improved by increasing the size of the swarm without the need to redefine or reprogram the system.

As a SI system, SR must abide by the conditions discussed in the previous section. In addition, the agents must be autonomous robots operating in the physical world with the ability to sense and actuate in a real environment. These simple robots must possess individual capabilities that are limited relative to the task to be carried out at the system level. In other words, they must be unable to solve the problem absent collaboration. This simplicity carries the added benefit that the robots are inexpensive and less prone to failure.

SR aims to study how a large number of simple robots can be designed so that a desired collective behaviour emerges from the local interactions among themselves and with the environment [73]. This research is not concerned with the design of the physical robots but of their self-coordination mechanisms so that they self-organise to develop the ability to fight fires autonomously and collaboratively. The design of the rules for the local interactions to lead to the emergence of the desired system's behaviour is not a trivial task. Therefore, the observation of desirable collective animal behaviour and the identification of the basic underlying self-coordination mechanisms can serve as sources of inspiration for the design of SR systems.

2.4. Particle Swarms Algorithm

Particle Swarm Optimisation (PSO) is one of the most successful SI algorithms, originally developed as a model of social behaviour inspired by earlier bird-flock simulations framed within the field of social psychology. In particular, Reynolds' boids [66] and Heppner and Grenander's artificial birds [33] strongly influenced the early developments. It was also influenced by experiments and theories in social psychology such as Sherif's experiments, Bandura's no-trial learning, and Latané's Social Impact Theory [47]. Therefore, the method is closely related to other simulations of social processes and experimental studies in social psychology whilst also having strong roots in optimisation and AI as well as applications in SR.

PSO is a global optimiser in the sense that it is able to escape poor suboptimal attractors thanks to a parallel search carried out by a swarm of cooperative particles. Its overall behaviour results from a combination of each particle's individual behaviour and the social behaviour that emerges from their interactions. The individual behaviour materialises as the trajectory of a particle pulled by an attractor. In most versions of the algorithm, this attractor results from some stochastic weighted average of an individual attractor (a particle's best experience) and a social attractor (best experience of neighbouring particles). The social behaviour is governed by how individually acquired information is propagated throughout the swarm. The individual and social behaviours are linked by the update of the social attractor in the trajectory equation. While the social behaviour is controlled by the neighbourhood structure (i.e. the sociometry), the individual behaviour is controlled by the settings of the coefficients in the trajectory equation.

2.4.1. Trajectory Difference Equation

The *Classical PSO* (CPSO) algorithm proposed in [76] is rearranged in Eq. (1) as a single 2nd order difference equation:

$$\begin{aligned} x_{ij}^{(t)} = x_{ij}^{(t-1)} &+ \omega_{ij}^{(t)} \left(x_{ij}^{(t-1)} - x_{ij}^{(t-2)} \right) \\ &+ iw_{ij}^{(t)} U_{(0,1)} \left(xb_{ij}^{(t-1)} - x_{ij}^{(t-1)} \right) \\ &+ sw_{ij}^{(t)} U_{(0,1)} \left(xb_{kj}^{(t-1)} - x_{ij}^{(t-1)} \right) \end{aligned} \tag{1}$$

where $x_{ij}^{(t)}$ is the coordinate j of the *position* of particle i at time-step t; $xb_{ij}^{(t)}$ is the coordinate j of the *best experience of particle i* by time-step t; k is the index identifying the particle whose best experience is the best in the neighbourhood at time-step t; $\omega_{ij}^{(t)}$, $iw_{ij}^{(t)}$, $sw_{ij}^{(t)}$ are the j^{th} components of the *inertia*, *individuality* and *sociality weights*, respectively, of particle i at time-step t; and $U_{(0,1)}$ is a random number from a uniform distribution in the range [0,1] resampled anew every time it is referenced.

In the original formulation [46], $\omega = 1$ and $iw = sw = 2$. This leads to an unstable system, as particles tend to diverge from their attractor(s). The first strategy used to control this so-called *explosion* consisted of bounding the size of each component of a particle's displacement, which helps prevent the explosion but does not ensure convergence or a fine-grain search. For CPSO, the coefficients in Eq. (1) can be set so that they ensure convergence and favour fine-grain search.

Alternatively, Clerc et al. [21] analysed the trajectory of a deterministic particle in the original PSO ($\omega = 1$) and developed so-called constriction factors (χ) that ensure convergence. This formulation is referred to as *Constricted PSO* (CoPSO).

For formulations including both inertia weight (ω) and constriction factor (χ), refer to [90, 13, 44].

2.4.2. Neighbourhood Structure

The neighbourhood structure in the original algorithm is the global topology, where every particle is connected to every other in the swarm, therefore having access at any time to the best solution found so far by any particle. This may lead to a rapid loss of diversity (implosion), which in turn may result in premature convergence to a poor suboptimal solution. While this can be controlled to some extent by the settings of the coefficients in the trajectory equation, numerous neighbourhood topologies have been proposed by reducing connectivity thus delaying the propagation of information throughout the swarm. A few neighbourhood topologies are shown in Fig. 2.

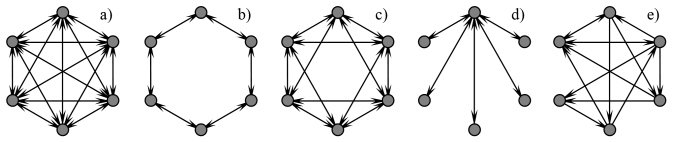


Figure 2: a) global topology; b) ring topology with two neighbours; c) ring topology with four neighbours; d) wheel topology; e) random topology.

An alternative to traditional topological neighbourhoods is to define the nearest neighbours based on actual Euclidean distance, or using the so-called *speciation* [50, 51, 62] where the local attractor (i.e. the best experience in the neighbourhood) is chosen as the best within adaptively generated species (clusters). For further studies on neighbourhood structures and inter-particle communication, refer to [28, 17, 53, 86].

2.4.3. Dynamic Environments

During the execution of a PSO run, the canonical algorithm has no means to detect changes in the location of the optima as particles are under the influence of outdated memories. Furthermore, as the search progresses and the swarm converges, diversity is progressively lost rendering the swarm unable to explore the dynamic environment to track the moving optima.

Carlisle et al. [14] adapted the algorithm to dynamic environments –specifically for tracking a moving beacon– by having particles periodically replace their memories with their currently held information, thus effectively forgetting experiences which are likely to have become obsolete.

Blackwell et al. [9] proposed using charged particles in environments where the optimum location changes randomly and with high severity. The use of charged particles aims to maintain diversity, thus enabling the swarm to better respond to such changes. The drawback is that this method requires the calculation of repulsive forces among all charged particles (n) with complexity of $O(n^2)$. Blackwell et al. [8] proposed splitting the population into a set of swarms that interact locally by an exclusion parameter and globally through an anti-convergence operator. Each swarm maintains diversity either by using charged or so-called quantum particles.

Parrot et al. [62] proposed a species-based PSO algorithm with a crowding prevention mechanism to locate multiple optima, coupled with the re-evaluation of each particle's memorised best location (xb) to track the changes in the dynamic environment. Yazdani et al. [88] proposed a *finder-trackers* multi-swarm algorithm combined with a mechanism to increase diversity based on a change in velocity vector and particle positions, a test point to detect change, a local search algorithm to enhance exploitation, and a so-called *awakening-sleeping* mechanism to improve efficiency. Sadeghi et al. [72] incorporated a memory of the locations of previous optima and a diversity mechanism based on k-means clustering to track changing optima.

A comprehensive comparison of different techniques to be incorporated to the PSO algorithm to handle dynamic environments can be found in [49, 88], including diversity-maintaining schemes, memory-based schemes, and multi-swarm schemes. For further reading, refer to [67, 63, 48].

3. Efficient Physics-Based Model of Wildfire Propagation

The use of aerial swarms to solve real-world problems has been increasing steadily, underpinned by technology advancements and falling prices. A key enabling technology is the development of self organisation algorithms that allow drones to communicate and allocate tasks amongst themselves, plan their trajectories, and self-coordinate their flights to achieve the swarm's objectives [20]. The aim of this research is to demonstrate the feasibility and potential of employing aerial swarm robotics to fight wildfires autonomously. In order to evaluate the collective firefighting behaviour that emerges from the proposed self-coordination mechanisms, an efficient physics-based model of fire propagation is required. A preliminary proof-of-concept of this research was presented in [42].

The fire propagation model developed is governed by a two-dimensional reaction-diffusion equation that describes the combustion of a mono-phase medium composed of pre-mixed gas of fuel and air. This comprises a simplified version of the model in [26], where the slope effect and water vapour pressure are

not considered. With this dimensionality reduction, some important phenomena in fire dynamics such as buoyancy are disregarded. It is assumed that there is no atmospheric wind, and mass transfer is neglected. In order to compensate for this, the diffusion coefficient is augmented and two terms are added to the energy balance equation to account for the energy losses due to convection and radiation in the vertical direction. Horizontal radiation is modelled to affect only neighbouring cells, since the cell-size of the discretised domain can be set larger than the optical thickness. The heat capacity at constant pressure of each chemical species is considered to be constant and equal to an average value within the considered temperature range, namely from 293 K to approximately 1,100 K. The mixed gases are confined in their original location, neither diffused nor transported, as if the pyrolysis gasses burned exactly where they had been released. The irreversible chemical reaction in Eq. (2) represents the combustion of the pyrolysis gasses (e.g. methane) in the air, which is considered to be composed of oxygen, carbon dioxide, water vapour and nitrogen:

$$\theta_1 \mathrm{CH_4} + \theta_2 \mathrm{O_2} \rightarrow \theta_3 \mathrm{CO_2} + \theta_4 \mathrm{H_2O} \quad (\text{in air}: \quad \theta_5 \mathrm{N_2}) \tag{2}$$

The model is formulated in Eq. (3) as a system of five PDEs, one for the enthalpy balance and four for the chemical species formation (CO_2 and H_2O) or consumption (Fuel and O_2).

$$\begin{cases} \rho c_p \dfrac{\partial T}{\partial t} = \overbrace{-\rho h_c \dfrac{M}{M_{fuel}} r}^{\text{Combustion}} \\ \qquad + \overbrace{\kappa \dfrac{\partial}{\partial x}\left(\dfrac{1}{c_p}\dfrac{\partial \left(c_p T\right)}{\partial x}\right) + \kappa \dfrac{\partial}{\partial y}\left(\dfrac{1}{c_p}\dfrac{\partial \left(c_p T\right)}{\partial y}\right)}^{\text{Thermal Diffusion}} \\ \qquad + \overbrace{\kappa \dfrac{\partial}{\partial x}\left(\dfrac{1}{c_p}\dfrac{\partial h_c}{\partial x}\right) + \kappa \dfrac{\partial}{\partial y}\left(\dfrac{1}{c_p}\dfrac{\partial h_c}{\partial y}\right)}^{\text{Enthalpy Diffusion}} \\ \qquad + \overbrace{C_a \left(T_{\text{amb}} - T\right)}^{\text{Vertical Convection}} \\ \qquad + \overbrace{\sigma\varepsilon\left[4\dfrac{\partial}{\partial x}\left(T^3 \dfrac{\partial T}{\partial x}\right)\delta x + 4\dfrac{\partial}{\partial y}\left(T^3 \dfrac{\partial T}{\partial y}\right)\delta y\right]}^{\text{2D Radiation}} \\ \qquad + \overbrace{\sigma\varepsilon\left[\dfrac{T_{\text{amb}}^4 - T^4}{\delta z}\right]}^{\text{Vertical Radiation}} - \overbrace{\rho c_p u_y \dfrac{\partial T}{\partial y} - \rho c_p u_x \dfrac{\partial T}{\partial x}}^{\text{Transport}} \\ \dfrac{\partial X_i}{\partial t} = -\dfrac{\theta_i}{\theta_{fuel}} \dfrac{M}{M_{fuel}} r \qquad \text{with } i = 1, ..., 4 \end{cases} \tag{3}$$

The system is closed with Eqs. (4) for the molar mass of the mixture, (5) for the heat capacity of the mixture, (6) for the combustion rate and (7) for the combustion enthalpy.

$$M = \sum_{i=1}^{5} X_i M_i \tag{4}$$

$$c_p = \sum_{i=1}^{5} X_i \frac{M_i}{M} c_{pi} \tag{5}$$

$$r\left(T, X_1, X_2\right) = -\delta^{+}_{(T,X_{1,2})} A_r T X_1^{0.5} X_2 \exp\left(-\frac{T_\mathrm{a}}{T}\right) \tag{6}$$

$$\begin{aligned} h_c &= \frac{H_c\left(T\right)}{M} = -\frac{1}{M}\sum_{i=1}^{5} \theta_i H_i\left(T\right) \\ &= \frac{1}{M}\sum_{i=1}^{5} \theta_i \left(H_{i,\text{ref}} + M_i c_{pi}\left(T_{\text{ref}} - T\right)\right) \end{aligned} \tag{7}$$

The molar mass of the mixture (M) in Eq. (4) is the summation of each chemical species molar mass (M_i) multiplied by the respective mass fraction (X_i). The averaged constant pressure heat capacity of the mixture (c_p) in Eq. (5) is obtained by weighted summation of the partial heat capacities (c_{pi}). The combustion rate (r) in Eq. (6) is the rate of fuel consumption, and follows the continuous exponential Arrhenius law. It is a function of temperature (T), fuel mass fraction (X_1) and oxygen mass fraction (X_2). The pre-exponential coefficient (A_r) and the activation temperature (T_a) are empirical parameters. The function $\delta^{+}_{(T,X_{1,2})}$ in Eq. (6) is a Kronecker delta as in Eq. (8).

$$\delta^{+}_{(T,X_{1,2})} = \begin{cases} 1 & \text{if} \quad T > T_\mathrm{p} \quad \text{and} \quad X_{1,2} > X_{1,2e} \\ 0 & \text{else} \end{cases} \tag{8}$$

This represents a simple extinction model: if the temperature is lower than the pyrolysis temperature (T_p), or if either the fuel mass fraction or the oxidant mass fraction is lower than the respective flame extinction value (X_{1e} or X_{2e}), combustion stops.

The combustion enthalpy in Eq. (7) consists of the summation of all formation enthalpies (H_i) at the specific local temperature (T). $H_{i,\text{ref}}$ and T_{ref} are reference values from tables.

According to our definitions and proposed classifications in Fig. 1, the model developed here is *continuous*, *deterministic* and *theoretical*. It is *physics-based*, with a few parameters to be calibrated (*light-grey-box*). In turn, the model would be classified as *physical* both in [64] and [80].

4. Self-Organising Swarm of Firefighting Drones

Since the PSO algorithm was inspired by models of decentralised flocks of birds, its potential to prescribe the desired behaviour of a swarm of self-organising drones is promising. A detailed study of PSO is beyond the scope of this paper. It suffices to say that the attractors in Eq. (1) can be combined into a single attractor at a given time-step (t) as follows [43, 41]:

$$\phi_{ij}^{(t)} = \iota_{ij}^{(t)} + \sigma_{ij}^{(t)} = iw_{ij}^{(t)}\, U_{(0,1)} + sw_{ij}^{(t)}\, U_{(0,1)} \tag{9}$$

$$p_{ij}^{(t)} = \frac{\iota_{ij}^{(t)}\, xb_{ij}^{(t)} + \sigma_{ij}^{(t)}\, xb_{kj}^{(t)}}{\phi_{ij}^{(t)}} \tag{10}$$

Clearly, the effect of the randomness incorporated into the coefficients in Eq. (1) is two-fold: it affects the trajectory of a particle towards the overall attractor (**p**), while also affecting the generation of this attractor as a stochastic convex combination of the individual ($\mathbf{xb}_i$) and the social ($\mathbf{xb}_k$) attractors. Different from classical formulations, these two features of the algorithm are decoupled here. Thus, the attractor is generated at every time-step from a uniform distribution within the rectangle generated such that its edges are parallel to the coordinate axes and it contains the current individual and social attractors as vertices. Once the attractor **p** is generated, the trajectory difference equation is as in Eq. (11).

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + \omega_{ij}^{(t)}\left(x_{ij}^{(t)} - x_{ij}^{(t-1)}\right) + \phi_{ij}^{(t)}\left(p_{ij}^{(t)} - x_{ij}^{(t)}\right) \tag{11}$$

The random variable ϕ can be realised as in Eq. (9) so that the probability distribution is triangular or trapezoidal as in the classical formulation, or a different density function can be chosen. Analysing the trajectory of a single particle with stationary attractor and constant coefficients, the settings of ω and ϕ control the type of behaviour (see Fig. 3). For optimisation purposes, convergent high-frequency harmonic oscillations are generally preferred, as there is no cost attached to large displacements. Conversely, low-frequency harmonic oscillations and smaller displacements are favoured for swarm robotic systems. Here we adopt the settings in Eq. (12), which ensure convergent low-frequency harmonic oscillations, with ϕ randomly generated from a uniform distribution within this range.

$$\begin{cases} \omega \in (0,1) \\ \phi \in \left[\left(\sqrt{\omega}-1\right)^2, (\omega+1)\right] \end{cases} \tag{12}$$

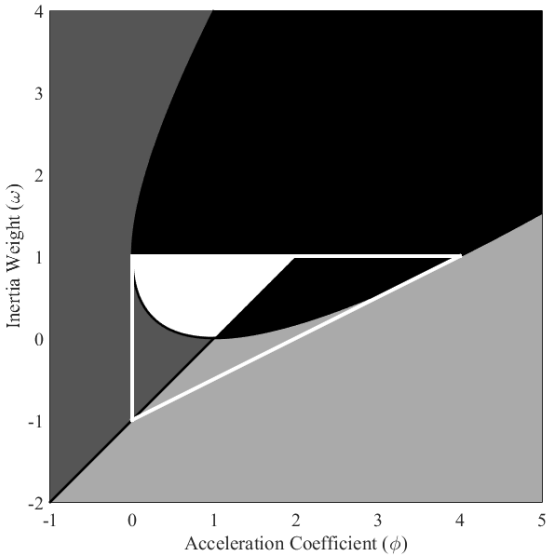


Figure 3: Regions in $\phi - \omega$ plane for different types of behaviour of a deterministic particle in PSO. Settings inside the white triangle ensure convergence, whilst settings in the black region inside the rotated parabola lead to harmonic oscillations. The superposed white subset of the black region corresponds to Eq. (12) and leads to low-frequency oscillatory behaviour.

A study of the influence of the coefficients in the trajectory difference equation on the types of behaviour of a particle is beyond the scope of this paper. Nonetheless, the convergence region and the equation of the parabola bounding the harmonic oscillatory behaviour in Fig. 3 (black region) can be found in [43], and the right boundary of the white region happens to be the same as ϕ_{mean} for *Behaviour Type 1* in Eq. (20) in [41].

While local neighbourhood topologies and other types of sociometries like distance-based nearest neighbours and speciation need to be investigated, the global neighbourhood topology is adopted for the initial studies presented in this paper.

As discussed in section 2.4.3, classical PSO requires some adaptations to cope with dynamic environments by addressing the problems of *outdated memory* and *diversity loss* [48]. The propagation of fire leads to a dynamic environment with multiple hotspots frequently and severely changing location and intensity. This feature is reinforced by drones attempting to suppress the fire and therefore modifying the environment and adding a form of *stigmergy* to their direct communication.

One technique to deal with dynamic environments is to re-evaluate the particles' memories when a change is detected, expected, or at regular intervals. This is not feasible in this case, as the sensors of physical drones are unable to acquire data from distant locations. An alternative is to replace the particles' memories with the currently held information. However, as the search progresses and the swarm converges, this leads to a fast loss of diversity rendering the swarm unable to explore the new landscape. Another technique is to re-initialise the particles' positions in the search-space, but it is impossible for physical drones to be instantly repositioned. Therefore, both the *outdated memory* and the *diversity loss* problems are addressed in this paper by erasing the individual memory of a drone when it has not been updated in the preceding 10 seconds, and then randomly re-initialising the memorised position. Thus, outdated memories are deleted and diversity is increased smoothly by new memories pulling the drones outwards via the trajectory equation rather than by an instant relocation. These new memorised positions are assigned temperatures one degree below ignition to avoid the search to continue once the fire is believed to have been suppressed.

The firefighting strategy is rather rudimentary at this stage: each drone simply searches for hotspots in the field, and every time it finds a location that is hotter than the hottest one stored in its individual memory, a third of its total water payload is dropped. After three drops, it must go back to the water source to replenish. Likewise, once a drone has travelled its total flying range, it must go back to its recharging docking station.

Fig. 4 provides a high-level description of the proposed self-organising swarm of firefighting drones coupled with the developed physics-based model of fire propagation. This flowchart is colour-coded for clarity, identifying five main sections and providing explanatory text-boxes that briefly describe the main tasks performed within each.

4.1. Initialisation

The settings for the physics-based fire propagation model and those for the agent-based model of the swarm of drones are

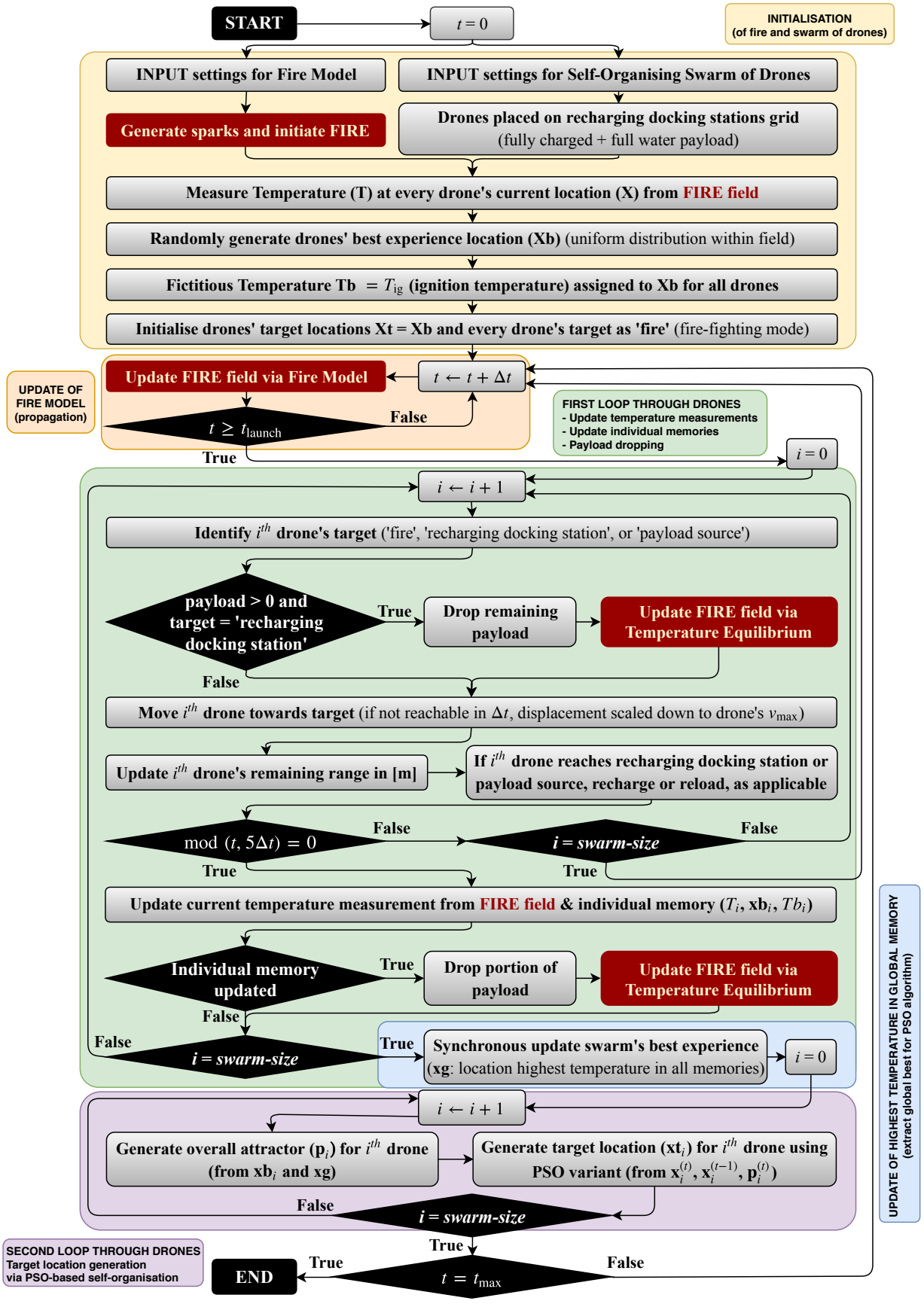


Figure 4: Flowchart with a high-level description of the model implementation for the proposed self-organising swarms of firefighting drones.

entered independently, defining both the wildfire scene and the firefighting tool. One or more sparks are randomly generated to initiate the fire, and the drones are placed on their recharging docking stations fully charged and fully loaded with water at the beginning of the simulation ($t = 0$).

The variables for the PSO-based self-coordination mechanism are initialised for each drone. Namely, the temperature at its current location (T) is measured, its memory storing the location of its best experience (**xb**) is generated randomly from a uniform distribution within the field, and the temperature associated is fictitiously set to the ignition temperature ($Tb = T_{\text{ig}}$). Every drone is set to firefighting mode, and its target location is set the same as the individual memory (**xt** = **xb**).

4.2. Update of Fire Model

The 2D fire model is updated by numerically computing its evolution for one time increment of the simulation, therefore requiring the discretisation of the space and time domains (Δx, Δy, Δt). From the initial fire sparks until the swarm of drones is launched, the fire propagates unchecked. In a real scenario, automated wildfire detection technology would include stationary (ground-based) visual systems, ground-based sensors, manned and unmanned surveying aircrafts, and/or satellite monitoring. For the purpose of these studies, the fire is simply allowed to propagate unchecked for a given amount of time (t_{launch}) before the drones are informed that a fire has been detected within the region of interest (no precise location is given). Once the firefighting activities start, the drones may affect the temperature field between updates of the fire model.

4.3. First Loop through Drones

This loop is nested within the time loop simulating the evolution of the whole system from $t = 0$ to $t = t_{\text{max}}$, with a given time increment (Δt) between iterations.

For each drone, its current target is identified, which states whether it must proceed to fight the fire, move towards the docking recharging station, or move towards the water source for replenishment. When a drone decides it must move towards the recharging docking station, whatever water is left in the tank is dropped at its current location. In such a case, the fire field needs to be updated by computing the new temperatures at the nodes affected by the drop. It is considered that a drop only affects instantaneously the four nodes from the discretised space which are closest to the coordinates of the drop. Thus, coefficients of influence are calculated for each of these four nodes as shown in Eq. (13) and in Fig. 5.

$$A_{ij} = \frac{\Delta x - |x - x_j|}{\Delta x} \cdot \frac{\Delta y - |y - y_i|}{\Delta y} \tag{13}$$

The fraction of the mass of the water dropped that affects each node is given by its coefficient of influence. The temperature at each of the four nodes is updated by means of a mass-based weighted average between the current nodal temperature and the temperature of the water dropped, as in Eq. (14).

$$T_{ij} \leftarrow \frac{A_{ij}\, m_{\text{drop}}\, T_{\text{water}} + m_{\text{gas}}\, T_{ij}}{A_{i,j}\, m_{\text{drop}} + m_{\text{gas}}} \tag{14}$$

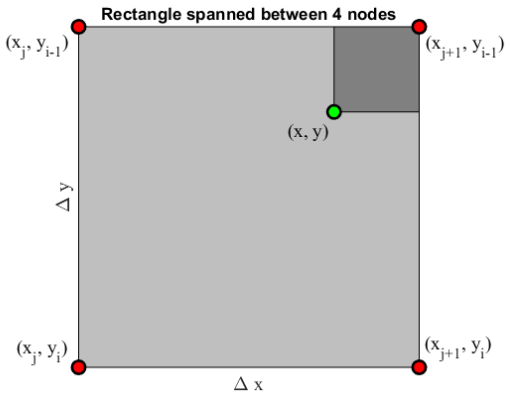


Figure 5: Rectangle spanned between four (red) nodes of the discretised space, with the green dot representing the location where a water payload is dropped. The coefficient of influence corresponding to the bottom-left node (x_j, y_i) is given by the ratio between the area of the dark-grey rectangle and that of the whole cell (i.e. $\Delta x \cdot \Delta y$).

The mass of the gas mixture of fuel and air (m_{gas}) is determined by the density of the gas (ρ) multiplied by the volume represented by a node: $\Delta x \cdot \Delta y \cdot z_{\text{th}}$, where z_{th} is the thickness of the gas layer provided as an input to the fire model.

Note that **A** is a 2×2 matrix that contains the four coefficients of influence for the nodes in question. The indices (i,j) are meant to facilitate the identification of the nodes they apply to. For instance, A_{21} applies to (x_j, y_i) in Fig. 5.

Whether the update of the fire field has been completed or unnecessary, the drone then moves towards its current target, whichever this may be, always ensuring that the maximum velocity permitted is not exceeded. Evidently, this physical displacement requires the update of the drone's remaining flying range. In this paper, the latter is simply measured in terms of the distance that can still be travelled before requiring a recharge. It goes without saying that the drone's payload is fully refilled and its flying range reset to the maximum once it reaches the water source and the recharging docking station, respectively.

In order to allow the fire to evolve between drones' interventions, measurements and memories can only be updated every five time-steps; i.e. for values of t which are multiples of 5 Δt. Thus, if $mod(t, 5\Delta t) = 0$, the drone measures the temperature at its current location. If this value is higher than the one in its memory, the latter is updated and a fraction of the water payload is dropped prompting an update of the fire field following the procedure described in Eqs. (13)–(14) and in Fig. 5.

The *outdated memory* and the *diversity loss* problems are addressed by erasing the individual memory of a drone when it has not been updated in the preceding 10 seconds, re-initialising its memorised best position randomly, and setting the associated temperature one degree below ignition.

4.4. Update of Highest Temperature in Collective Memory

Since the drones operate in parallel, it is more realistic to update the global memory in a synchronous manner once all individual memories are up-to-date. Therefore, it needs to be carried out outside the first loop through drones discussed in

the previous section. This update simply consists of extracting the location and temperature of the hottest spot from the current memories of all drones.

4.5. Second Loop through Drones

The choice of a synchronous update of the global memory makes this loop necessary. Otherwise, both the asynchronous update of the swarm's best experience and the tasks described below could be performed at the end of the first loop through the drones discussed in section 4.3. Hence the two loops through the drones are sequential and nested within the loop through time that simulates the evolution of the whole system. Note that the second loop is only executed every $5\Delta t$.

As discussed before, the PSO formulation proposed here decouples the generation of the overall attractor and the evaluation of the trajectory difference equation. Thus, the overall attractor for drone i at time t ($\mathbf{p}_i^{(t)}$) is stochastically generated from a uniform distribution within the rectangle spanned between the location of the individual best experience ($\mathbf{xb}_i^{(t)}$) and that of the global best experience ($\mathbf{xb}_k^{(t)}$) as in Eq. (15). Sub-index k identifies the drone holding the global best experience.

$$p_{ij}^{(t)} = xb_{ij}^{(t)} + U_{(0,1)}\left(xb_{kj}^{(t)} - xb_{ij}^{(t)}\right) \tag{15}$$

The acceleration coefficient ($\phi_{ij}^{(t)}$) is generated independently from the overall attractor as in Eq. (16). This is different from the original coupled formulations in Eqs. (9)–(10). In addition, the pair (ϕ, ω) is kept within the white region in Fig. 3 to favour low-frequency harmonic oscillatory behaviour.

$$\phi_{ij}^{(t)} = \left(\sqrt{\omega} - 1\right)^2 + U_{(0,1)}\left[(\omega + 1) - \left(\sqrt{\omega} - 1\right)^2\right] \tag{16}$$

The target location for drone i for the next iteration ($\mathbf{xt}_i^{(t+1)}$) is generated as in Eq. (17). While the resemblance to the PSO trajectory equation is evident, this is not really a difference equation because it returns $\mathbf{xt}^{(t+1)}$ as a function of $\mathbf{x}^{(t)}$ and $\mathbf{x}^{(t-1)}$. This is the drone's target when in firefighting mode.

$$xt_{ij}^{(t+1)} = x_{ij}^{(t)} + \omega_{ij}^{(t)}\left(x_{ij}^{(t)} - x_{ij}^{(t-1)}\right) + \phi_{ij}^{(t)}\left(p_{ij}^{(t)} - x_{ij}^{(t)}\right) \tag{17}$$

5. Numerical Experiments

The domain in these experiments is a field of 100x100 m^2 with fuel uniformly distributed except for a small band of 9.6 m around the field so that combustion does not take place near the boundaries. Thus, the total area covered with fuel equals 6,529 m^2 whilst the total fuel energy available is 218.30 GJ. The adopted fuel is short grass, referred to as Fuel Model 1 in [3]. No-wind condition is assumed, and all experiments are run for 480 s. The remaining settings are shown in Table 1.

The fire propagation model is implemented using an explicit numerical scheme consisting of a 2$^{\text{nd}}$ order Finite Difference Method (FDM) in space with Dirichlet boundary conditions and a 4$^{\text{th}}$ order Runge-Kutta (RK4) method for the integration in time. Implicit methods are impractical for our needs.

Experimental results, figures and animations are available open access via the *figshare* online digital repository.

5.1. Fire Model Calibration Experiment

In order to make the numerical experiments as realistic as possible to test the proposed firefighting system, the physics-based fire spread model developed is calibrated against predictions made by the commercial simulator FARSITE [27]. The latter cannot be used directly because it does not comply with the requirement of being physics-based to interact with the fire suppressant. Nonetheless, FARSITE is useful for calibration purposes because it has been validated against historical fires.

The calibration simulations are performed with a single-point ignition of a uniformly distributed fuel bed. The objective is to minimise the error between the area affected by the fire predicted by our model (A_{a}) and the one predicted by FARSITE (A_{ref}). Three variables are chosen for calibration, namely the thermal diffusion coefficient (κ), Arrhenius pre-exponential coefficient (A_{r}), and the horizontal optimal thickness ($\delta x = \delta y$). The optimisation problem is formulated in Eq. (18), where $T_{\text{ref}} = 1{,}200$ K is the temperature of wood combustion in the absence of wind and crown fire. The second constraint ensures that the horizontal optical thickness is smaller than the mesh size, which is a model assumption for the local radiation.

$$\begin{aligned} \underset{[\kappa, A_{\text{r}}, \delta_{\text{x}}]}{\text{Minimise}} \quad & (A_{\text{a}} - A_{\text{ref}})^2 \\ \text{Subject to} \quad & T_{\max} = T_{\text{ref}} \\ & \delta_x < \Delta_x \end{aligned} \tag{18}$$

The optimisation problem is solved at $t = 600$ s using SQP, which returns the following values of the calibration variables: $\kappa = 0.988$ W·m^{-1}·K^{-1}, $A_{\text{r}} = 5.667{\times}10^{-5}$, and $\delta x = \delta y = 0.112$ m. As expected, κ adopts an augmented value to compensate for the lack of horizontal transport. The agreement between the fire front predicted by our calibrated model and the one predicted by FARSITE is shown in Fig. 6 and in [dataset] [39].

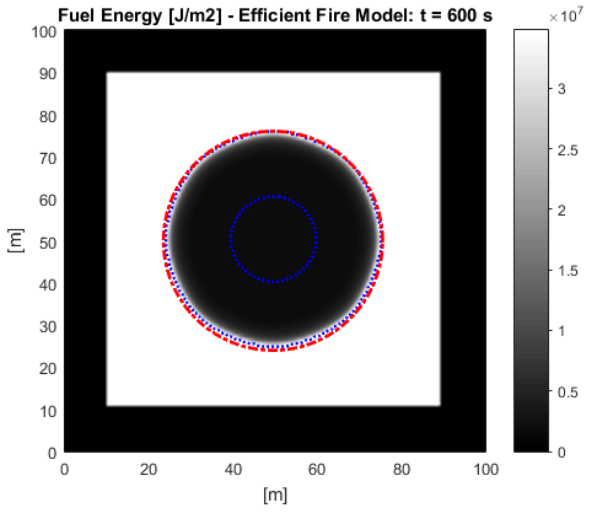


Figure 6: Fuel energy field 600 s after ignition during the calibration experiment. The red line depicts FARSITE's fire front, the outer blue line shows our model's fire front, the dark region inside shows the burnt area, and the inner blue line encloses the burnt region with current temperature below ignition.

Table 1: General settings for the numerical experiments, including the three calibrated variables.

R	Universal gas constant	8.3140	$\mathrm{J{\cdot}mol^{-1}{\cdot}K^{-1}}$
T_{amb}	Ambient temperature	294.1500	K
T_{ref}	Reference temperature (for h and c_{p} calculation)	298.1500	K
P_{ref}	Reference pressure	101325	Pa
P_{amb}	Ambient pressure	101325	Pa
T_{ig}	Ignition temperature	370	K
ρ	Gas mixture density	1.2172	$\mathrm{kg{\cdot}m^{-3}}$
z_{th}	Thickness of layer of fuel (gas mixture)	1.00	m
A_{r}	Arrhenius pre-exponential coefficient (**calibrated**)	5.667×10^{-5}	-
C_{a}	Turbulent convection coefficient in atmosphere	0.0600	$\mathrm{J{\cdot}m^{-3}{\cdot}K^{-1}{\cdot}s^{-1}}$
κ	Thermal diffusion coefficient (**calibrated**)	0.988	$\mathrm{W{\cdot}m^{-1}{\cdot}K^{-1}}$
c_{p}	Specific heat capacity at constant pressure	calculated	$\mathrm{J{\cdot}kg^{-1}{\cdot}K^{-1}}$
X_{f}	Initial fuel molar fraction	0.1000	-
σ	Stefan-Boltzmann constant	5.6704×10^{-8}	$\mathrm{W{\cdot}m^{-2}{\cdot}K^{-4}}$
ε	Emissivity factor (< 1)	0.70	-
$\delta x = \delta y$	Horizontal optical thickness (**calibrated**)	0.112	m
δz	Vertical optical thickness	1.0	m
h_{c}	Specific combustion enthalpy	calculated	$\mathrm{J{\cdot}kg^{-1}}$
$\Delta x = \Delta y$	Cell-size	0.80	m
Δt fire	Time between fire-spread updates	0.20	s
Δt drone	Time between sensor measurements and memory updates	1.00	s
t_{dep}	Time at which drones are deployed	20	s
Payload	Total amount of water carried by a drone (3 drops of 40 kg)	120	kg
Flying range	Maximum distance a drone can travel between recharges	2000	m
v_{max}	Maximum drone velocity permitted	10	$\mathrm{m{\cdot}s^{-1}}$

5.2. Fire Propagation Experiments

Two experiments are carried out to observe unchecked fire propagation, and to keep the resulting values of the area affected (A_{a}) and of the energy consumed (E_{c}) as frames of reference. The fire is started from a single spark (or source) in the centre of the field for the first experiment, whereas three additional sources are randomly generated for the second experiment (i.e. four sparks). The temperature and the fuel energy fields at $t = 10$ s and at $t = 480$ s are shown in Fig. 7 for the first experiment and in Fig. 8 for the second one. As the fires propagate, the evolution of the A_{a} and that of the E_{c} for each fire are shown in Fig. 9, with the final values provided in Table 2.

Table 2: A_{a} and E_{c} 480 s after the fire started. Total area with fuel is 6,529 $\mathrm{m^2}$ and total fuel energy before fire is 218.30 GJ.

No. of sources	A_{a}	E_{c}
1 fire source	1,313.92 $\mathrm{m^2}$	38.91 GJ
4 fire sources	2,938.88 $\mathrm{m^2}$	89.56 GJ

Results from the numerical experiments in this section are available in [dataset] [39], including animations of the evolutions of the temperature and of the fuel energy fields which appear sufficiently realistic to be used as a platform for initial testing of the proposed self-organising firefighting system.

5.3. Scalability Experiments

Once the model developed for fire propagation runs realistically and reasonably fast, it is coupled with the proposed model of self-organising swarms of firefighting drones as shown in Fig. 4. Recall that, since the focus is on the swarm-intelligent coordination mechanisms so that the drones self-organise to develop the ability to fight fires autonomously and collaboratively, the simulated drones are collision-free at this stage.

It is important to note that the model of the firefighting system is stochastic, and therefore multiple runs are necessary for each experiment in order to compute basic statistics required to derive reliable conclusions from the results. Even though these models are efficient, the overall coupled model in addition to the overheads for generating figures, animations, data output files, and organising them automatically in a folder structure, is about four to five times slower than real time in a standard desktop computer. Therefore, only a limited number of experiments can be carried out within a reasonable time frame.

For the remaining experiments in this paper, 31 independent runs are carried out for the statistics. Evidently, this is not statistically significant. In fact, Fig. 10 shows a plot of the mean A_{a} by the fire after an eight-minute propagation against the number of samples used for the mean calculations. Clearly, convergence has not been reached after 28 successful runs (three runs

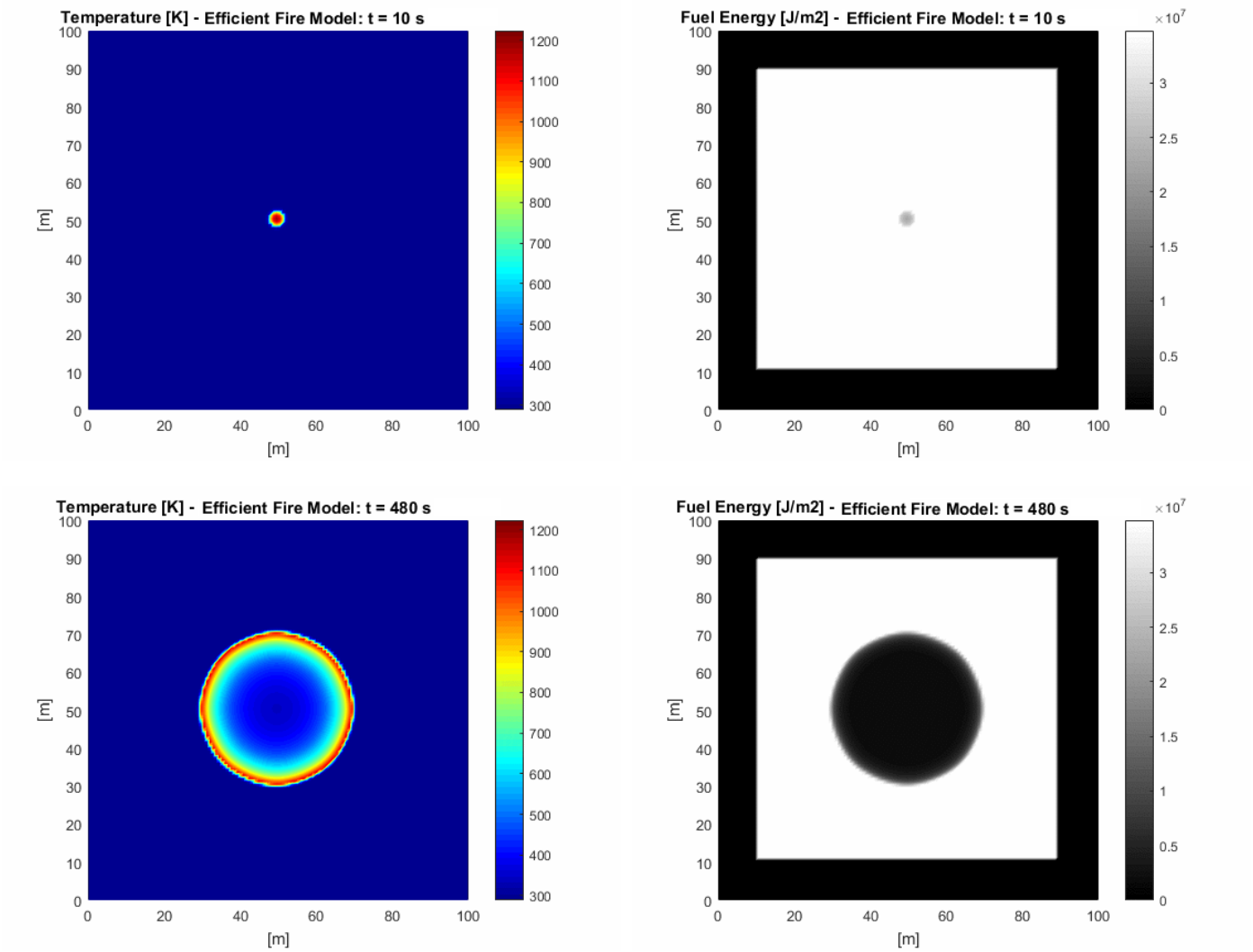


Figure 7: Single run of the fire model started from a single spark (i.e. one fire source). The figures on the left are snapshots of the temperature field at t = 10 s (top) and t = 480 s (bottom). The figures on the right are snapshots of the fuel energy field at t = 10 s (top) and t = 480 s (bottom).

failed to suppress the fire). Nonetheless, these runs still serve to capture general trends. For example, a single run of an experiment may show that the system fully suppresses a given wildfire whilst 31 independent runs of the same experiment show that success is an unlikely outcome. For more precise and definite conclusions, a higher number of runs must be carried out.

Since swarm-robotic systems are inherently scalable, a fire can always –in principle– be extinguished given enough time and a sufficiently high number of firefighting drones, even if this number is unrealistically high. Thus, the two fires from section 5.2 are considered again, but now coupled with a self-organising swarm of drones aimed at suppressing them. In each case, the experiments start with a swarm-size that appears insufficient. The number of drones is increased thereafter to analyse the impact on the success rate, the suppression time (t_{sup}), the area affected (A_{a}), and the energy consumed (E_{c}) by the fire.

The experimental results are provided in Table 3 and Fig. 11 for the fire started from a single source and swarm-sizes ranging from 30 to 70 drones, and in Table 4 and Fig. 12 for the fire started from four sources and swarm-sizes ranging from 70 to 110 drones. Output data files resulting from the numerical experiments in this section are available in [dataset] [40].

Since the curves of the mean A_{a} and of the mean E_{c} display convergent behaviour for successful runs (SRs) whereas they are monotonically increasing for unsuccessful runs (URs), they are considered separately in Figs. 11 and 12. Plotting mean curves with SRs and URs averaged together would provide little insight. However, this has the undesirable consequence that, in most cases, even less than 31 runs are considered in the mean calculations. Likewise, SRs and URs are also considered separately in Tables 3 and 4.

Starting from the single-source fire, Table 3 shows that a swarm of 30 drones is unable to consistently suppress this fire, failing to do so around 75% of the time. Furthermore, by analysing the animations in [dataset] [40], it can be observed that URs are indeed due to insufficient firefighting power. This can also be inferred from the curves for URs in Fig. 11 for 30 drones, which grow exponentially. Increasing the swarm-size

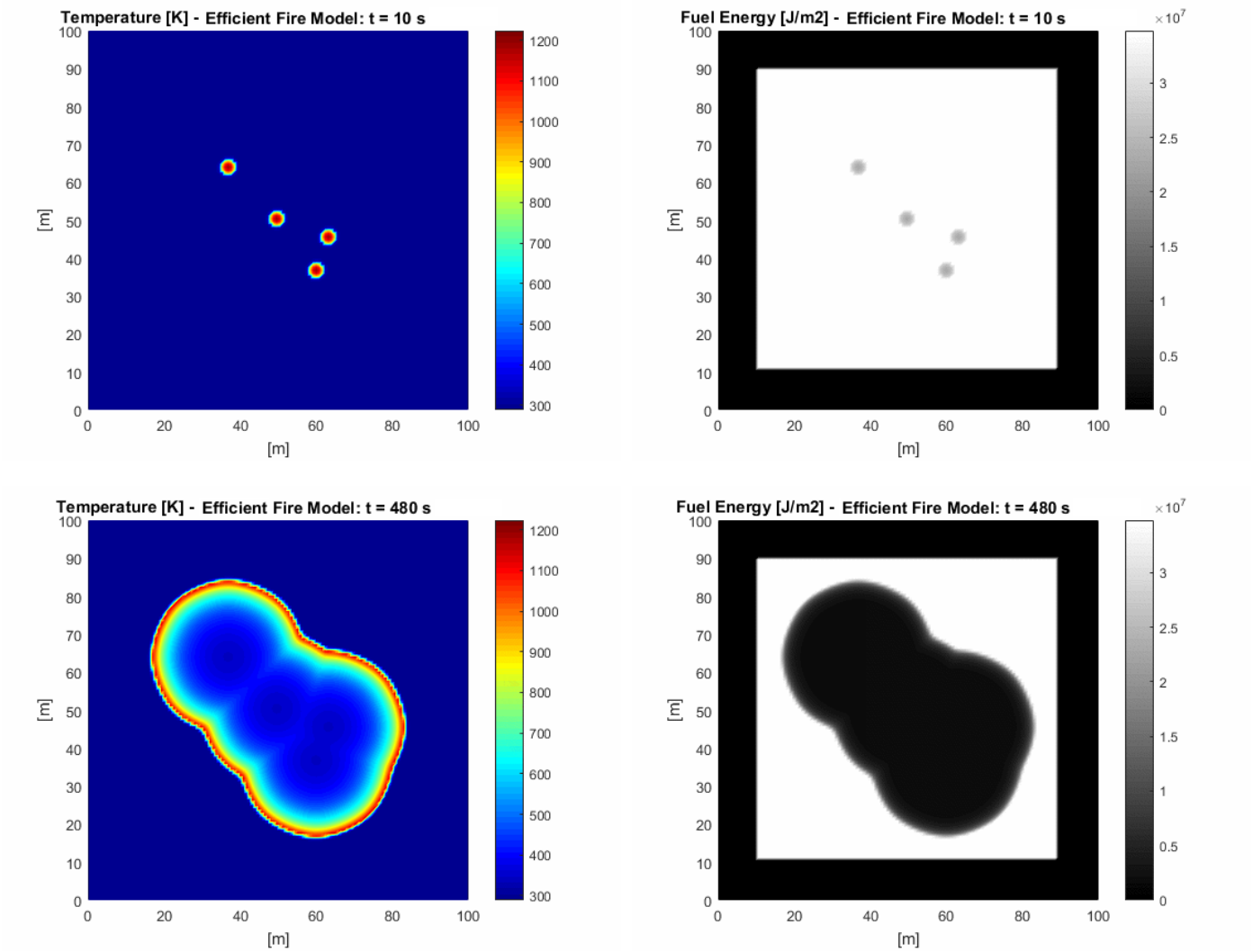


Figure 8: Single run of the fire model started from four sparks (i.e. four fire sources). The figures on the left are snapshots of the temperature field at t = 10 s (top) and t = 480 s (bottom). The figures on the right are snapshots of the fuel energy field at t = 10 s (top) and t = 480 s (bottom).

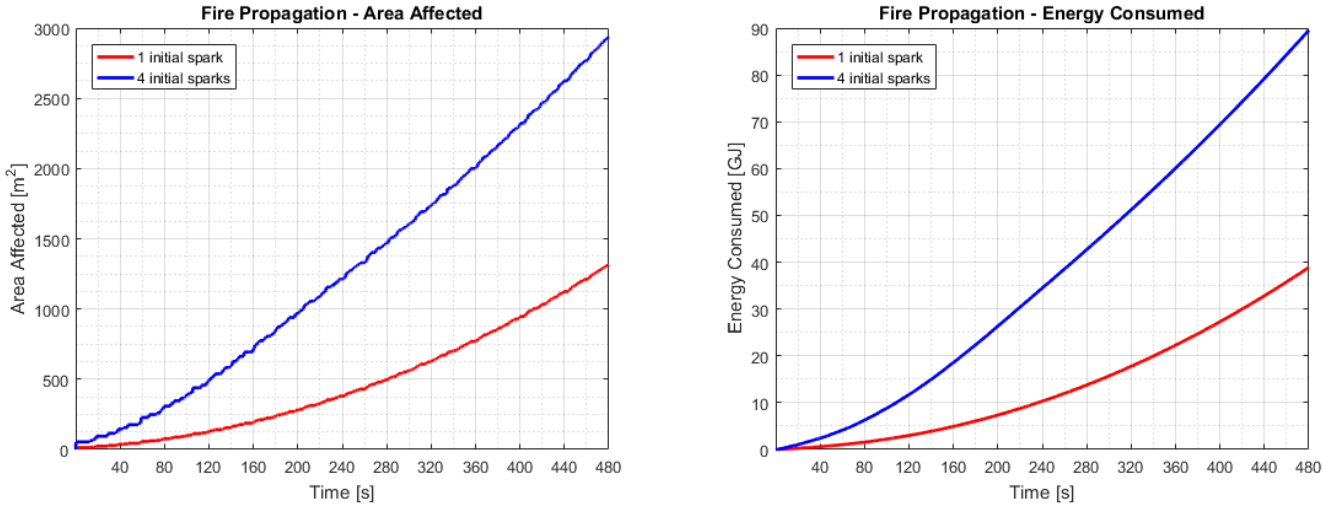


Figure 9: Area affected (A_a) [m^2] and energy consumed (E_c) [GJ] for a single run of the fire model after having propagated for 480 s. Results from two fires are shown, one which started from a single source (red) and another which started from four sources (blue).

Table 3: Information extracted 480 s after a fire was originated from a single source, with $\omega = 0.75$ and 31 runs carried out for each swarm-size for the statistics. Means (μ) and standard deviations (σ) are calculated separately for successful runs (SRs) and unsuccessful runs (URs). A fire is deemed suppressed when the swarm believes it to be so, and returns to the docking stations. Total area with fuel is 6,529 m^2 and total fuel energy before fire is 218.30 GJ.

m	SRs Actual	SRs Deemed	Mean t_{sup} [s]	Successful Runs (SRs)				Unsuccessful Runs (URs)			
				μ_{A_a}[m^2]	σ_{A_a}[m^2]	μ_{E_c} [GJ]	σ_{E_c} [GJ]	μ_{A_a}[m^2]	σ_{A_a}[m^2]	μ_{E_c} [GJ]	σ_{E_c} [GJ]
30	7/31	8/31	236.43	61.07	23.13	1.18	0.46	343.15	158.29	7.75	3.99
40	19/31	25/31	255.05	52.75	12.44	0.99	0.25	97.92	19.88	1.85	0.39
50	28/31	30/31	203.52	43.06	10.47	0.78	0.21	205.65	112.22	5.01	3.19
60	30/31	31/31	143.47	34.88	7.39	0.62	0.13	551.04	–	15.30	–
70	29/31	31/31	100.73	33.30	3.94	0.57	0.07	401.28	170.16	10.84	5.06

Table 4: Information extracted 480 s after a fire was originated from four sources, with $\omega = 0.75$ and 31 runs carried out for each swarm-size for the statistics. Means (μ) and standard deviations (σ) are calculated separately for successful runs (SRs) and unsuccessful runs (URs). A fire is deemed suppressed when the swarm believes it to be so, and returns to the docking stations. Total area with fuel is 6,529 m^2 and total fuel energy before fire is 218.30 GJ.

m	SRs Actual	SRs Deemed	Mean t_{sup} [s]	Successful Runs (SRs)				Unsuccessful Runs (URs)			
				μ_{A_a}[m^2]	σ_{A_a}[m^2]	μ_{E_c} [GJ]	σ_{E_c} [GJ]	μ_{A_a}[m^2]	σ_{A_a}[m^2]	μ_{E_c} [GJ]	σ_{E_c} [GJ]
70	0/31	1/31	–	–	–	–	–	959.42	280.12	23.04	7.68
80	8/31	8/31	357.18	278.48	64.75	5.56	1.45	539.38	215.38	11.85	5.23
90	22/31	25/31	335.42	247.83	52.29	4.89	1.17	376.75	123.90	7.97	2.84
100	28/31	28/31	263.91	214.38	39.63	4.12	0.82	341.55	25.29	6.72	0.59
110	31/31	31/31	232.00	191.19	34.29	3.63	0.76	–	–	–	–

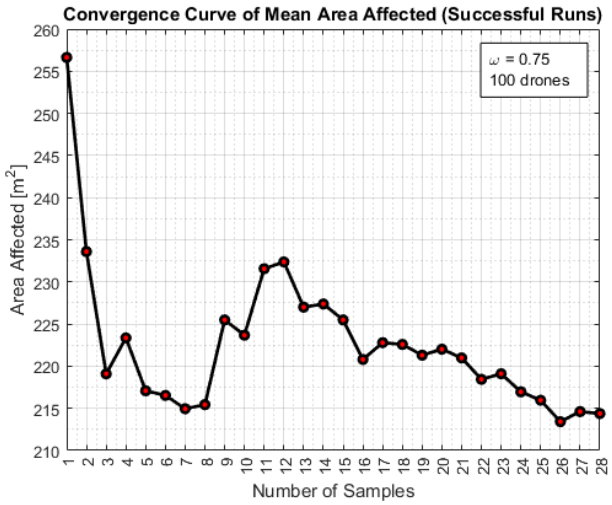


Figure 10: Mean A_a after 480 s fire propagation versus number of samples used for the statistics. The fire started from four sources, with 100 drones fighting its propagation. Convergence still not achieved for 28 samples.

to 40 already has a big impact, approximately tripling the number of SRs, both actual and deemed. Interestingly, the swarm believes to be successful about 80% of the time when in fact it only suppresses the fire about 60% of the time. Analysing the animations in [dataset] [40], it can be observed that the swarm is close to suppressing the fire in all URs. This can also be inferred from the curves for URs in Fig. 11 for 40 drones, which grow with decreasing slope. In the six cases in which the swarm returned to the docking station with the fire not fully extinguished, this happened near the end of the simulation and therefore the fire did not have time to grow back unchecked.

Further increasing the swarm-size to 50, 60 and 70, the actual success rate grows above 90%, with the swarm believing to have suppressed the fire every time but once. Analysing the animations in [dataset] [40] for these cases, it can be observed that the fire is almost extinguished in every UR. Therefore, the problem is not a lack of firefighting power. Being able to only measure temperature above their exact current positions, and only taking measurements once per second whilst $v_{\max} = 10$ m/s, the swarm is unable to fine-tune the search repeatedly overflying small remaining hotspots. This makes the swarm believe that the fire has been suppressed in a few cases, returning to the docking stations prematurely and letting the fire grow back unchecked. For instance, see Fig. 13 and its corresponding animation in [dataset] [40]. This is evident in the curves for URs in Fig. 11 for 50, 60 and 70 drones, which display inflection points after which the fire grows unchecked. Beware that the mean curves in Fig. 11 and the statistics in Table 3 for 50, 60 and 70 drones must be taken with a pinch of salt given the small number of URs involved. As to the SRs, the curves in Fig. 11 and the results in Table 3 display the expected trend: as the swarm-size (m) increases, the means and standard deviations of the suppression time (t_{sup}), the A_a and the E_c decrease.

The fire which started from four sources is evidently more challenging, and a 70-drone swarm now fails to suppress it every time (see Table 4). Analysing the animations in [dataset] [40], it is clear that there is a lack of firefighting power. An example of a failed run is shown in fig. 14. Scaling up the swarm to 80 drones starts showing some success, with the fire being

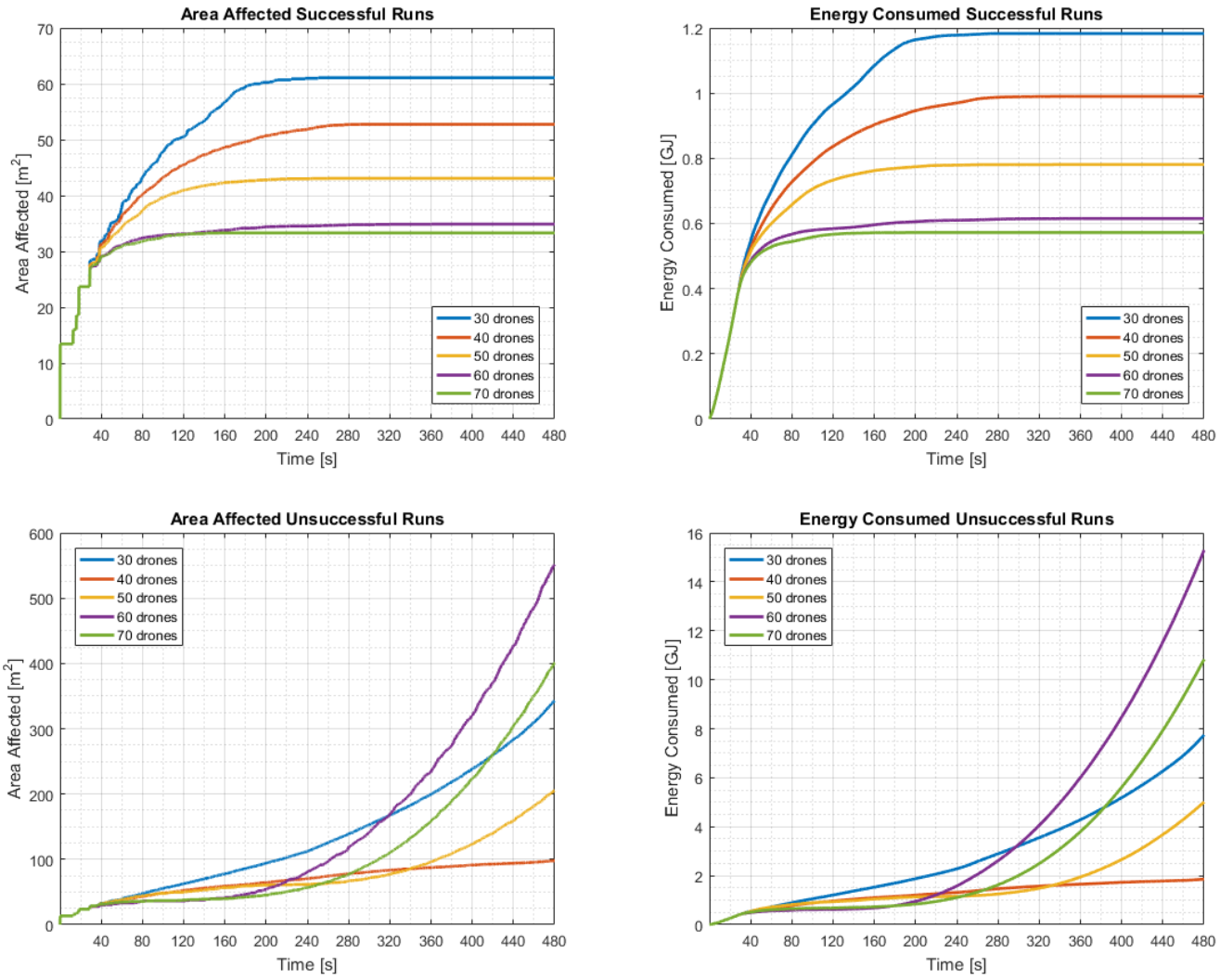


Figure 11: Mean A_a and mean E_c by a fire originated from a single source and propagated for 480 s corresponding to different swarms composed of 30 to 70 drones fighting its propagation (ω = 0.75). Whilst successful (SRs) and unsuccessful runs (URs) are considered separately, the mean values are calculated out of data gathered from 31 runs in total for each swarm-size. That is to say that the sum of the numbers of SRs and URs for a given swarm-size equals 31.

suppressed about 25% of the time. Analysing the animations in [dataset] [40] for the URs, it can be observed that the fire is contained sometimes even if not fully suppressed (e.g. [dataset] [40], 4 fire sources, Setting 2, Runs 7, 10 and 15), whereas other times it goes completely out of control (e.g. [dataset] [40], 4 fire sources, Setting 2, Runs 1, 2 and 3).

Further increasing the swarm-size has the desired effect. Success rate is about 70% for 90 drones, 90% for 100 drones and 100% for 110 drones. For 90 drones, the phenomenon of the swarm returning to the docking stations prematurely is still observed in 3/31 runs. Analysing the curves in Fig. 12 and the results in Table 4, the expected trend is observed: as the swarm-size (m) increases, the means and standard deviations of the t_{sup}, the A_a and the E_c decrease. Note that every run is unsuccessful for 70 drones whereas every run is successful for 110 drones.

It has been shown in these experiments that the proposed autonomous firefighting system is straightforwardly scalable, with the expected result that increasing swarm-sizes lead to increasing firefighting power. In turn, this implies increasing ability to cope with larger wildfires and to reduce their negative impact in terms of area affected (A_a) and energy consumed (E_c).

5.4. Control Coefficients Experiments

This set of experiments aims to provide some initial insight into the impact of the inertia weight on the firefighting performance of the system. Since Eq. (12) in our PSO formulation directly links the bounds of the acceleration coefficient (ϕ) to the inertia weight (ω), the latter is the only coefficient left to control the speed of convergence of the swarm.

For these experiments, a swarm of 100 drones is set to suppress a fire originated from four sources. Ten values of ω ranging from 0.50 to 0.95 are considered, with ϕ generated randomly from a uniform distribution as in Eq. (16). Results are shown in Table 5 and Fig. 15.

The output data files resulting from the numerical experiments in this section are available in [dataset] [37].

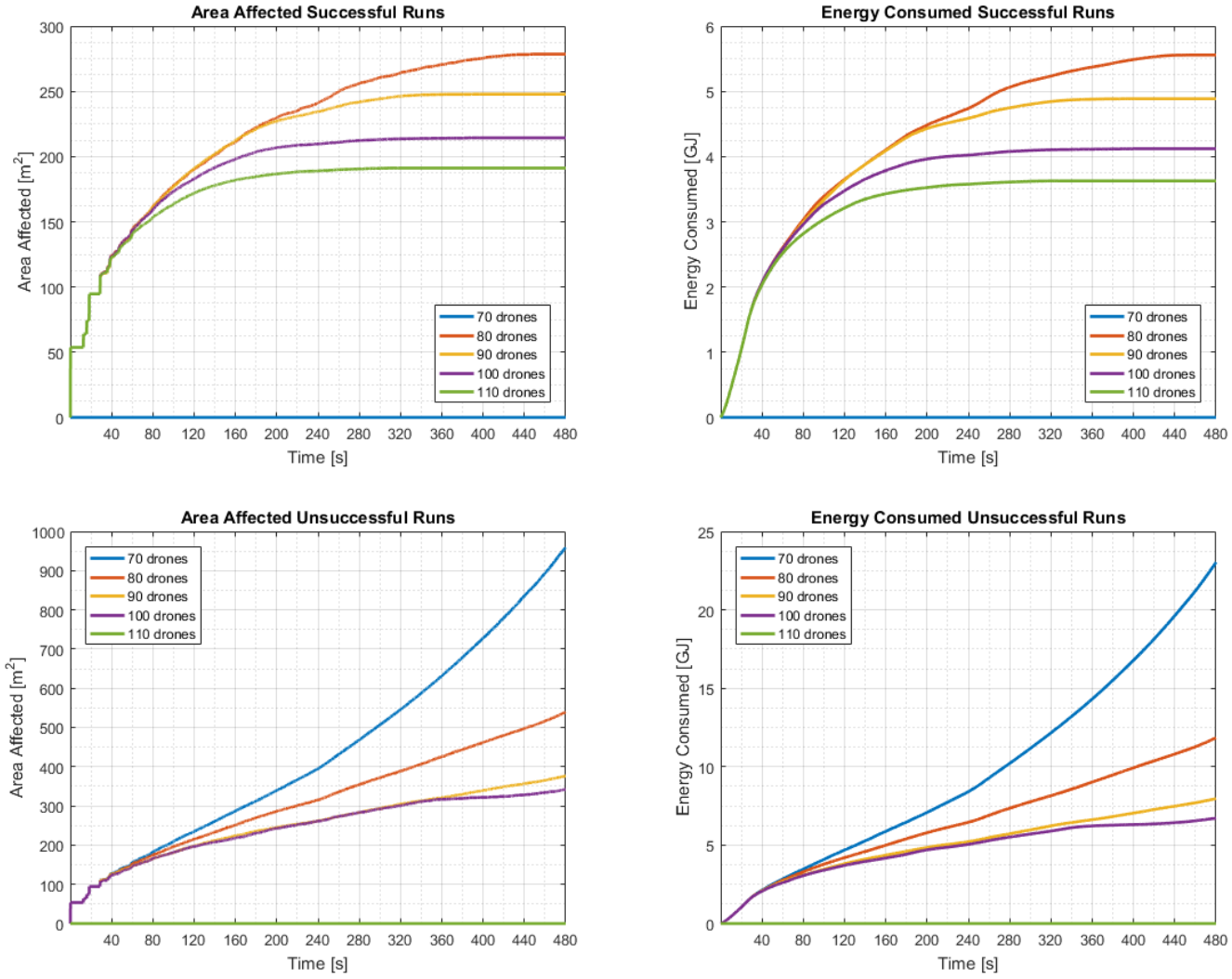


Figure 12: Mean A_a and mean E_c by a fire originated from four sources and propagated for 480 s corresponding to different swarms composed of 70 to 110 drones fighting its propagation (ω = 0.75). Whilst successful (SRs) and unsuccessful runs (URs) are considered separately, the mean values are calculated out of data gathered from 31 runs in total for each swarm-size. That is to say that the sum of the numbers of SRs and URs for a given swarm-size equals 31.

For this formulation and for a deterministic particle –i.e. drone– with a stationary attractor, $\omega \in (0, 1)$ guarantees convergence whilst ϕ within the range stated in Eq. (12) leads to low-frequency harmonic oscillations. Within this range and under these ideal conditions, the lower the ω the faster the convergence. Once randomness is re-introduced, note that a lower ω also implies a smaller stochastic range for ϕ (see Fig. 3).

As can be observed in Table 5, the two lowest values of ω result in the lowest success rates, as it is to be expected due to favouring fast convergence over exploration. For $\omega \geq 0.60$, success rate is above 84%, both actual and deemed. There are only a few cases of premature return to the docking stations, phenomenon that was discussed in previous experiments.

Although the pattern is not as clear as in the scalability experiments, it can be observed in Table 5 and Fig. 15 that performance of the proposed self-organising swarm of firefighting drones improves as ω increases. This may be due to the fact that larger values of ω favour exploration or to the larger range of values allowed for ϕ to be generated. In this sense, it may be interesting to run these experiments for a deterministic ϕ.

5.5. *Fault Tolerance Experiments*

Given that the proposed autonomous firefighting system is based on swarm robotics, scalability was expected at the outset and confirmed in the experiments carried out in section 5.3. Likewise, this decentralised, collaborative and self-organising system is expected to be able to cope with the failure or loss of a few drones without jeopardising the mission. In other words, it is predicted to be fault-tolerant.

In order to confirm this, an experiment is designed in this section in which one drone is lost at regular intervals during firefighting operations. Thus, a swarm initially composed of 100 drones with ω = 0.75 is set to suppress a fire originated from four sources during 480 s. The final number of drones will depend on how long the swarm takes to suppress the fire. The first drone is lost at t =30 s (recall that drones are launched

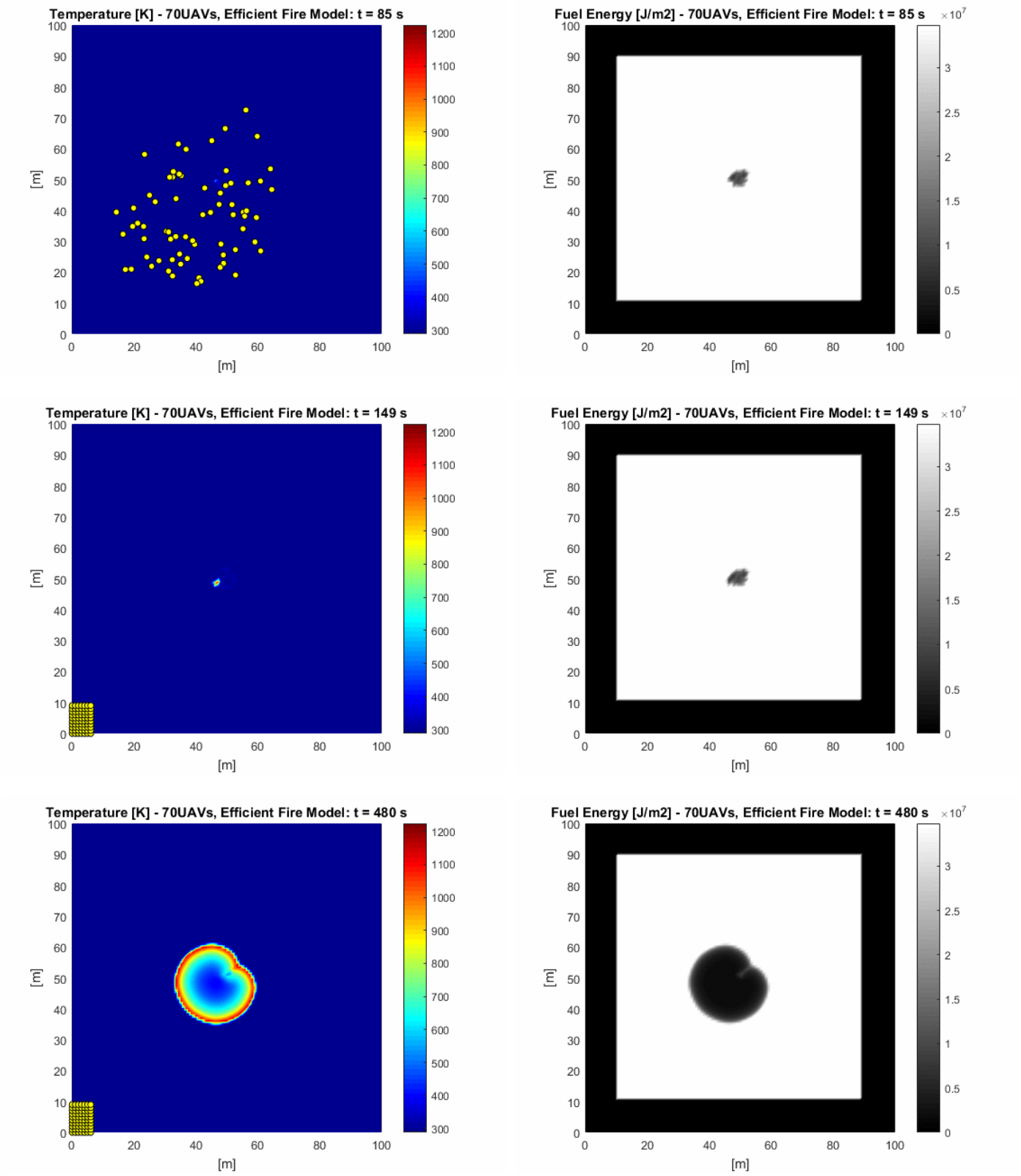


Figure 13: Swarm of 70 drones fighting a single-source fire propagating for 480 s. The figures on the left show the temperature fields at t = 85 s (top), t = 149 s (middle) and t = 480 s (bottom). The figures on the right show the corresponding fuel energy fields. By t = 149 s, the fire is almost suppressed and the swarm returns to the docking stations leaving the fire to propagate unchecked. The corresponding animation is available in [dataset] [40], 1 fire source, Setting 5, Run 12.

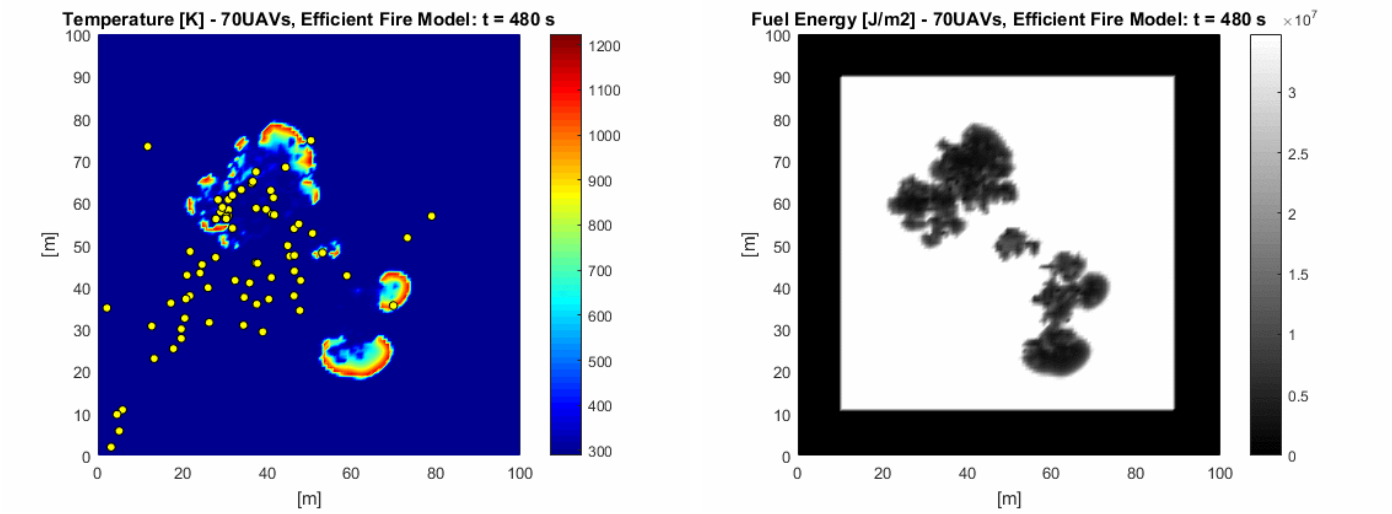


Figure 14: Example of a run for a 70-drone swarm failing to suppress a four-source fire. The figure on the left shows the temperature field while the one on the right shows the fuel energy field at the end of the simulation (t = 480 s).

Table 5: Information extracted 480 s after a fire was originated from four sources, with a swarm of 100 drones fighting its propagation and 31 runs carried out for each setting of the inertia weight (ω) for the statistics. Means (μ) and standard deviations (σ) are calculated separately for successful runs (SRs) and unsuccessful runs (URs). A fire is deemed suppressed when the swarm believes it to be so, and returns to the docking stations. Total area with fuel is 6,529 m^2 and total fuel energy before fire is 218.30 GJ.

ω	SRs Actual	SRs Deemed	Mean t_{sup} [s]	Successful Runs (SRs)				Unsuccessful Runs (URs)			
				μ_{A_a}[m^2]	σ_{A_a}[m^2]	μ_{E_c} [GJ]	σ_{E_c} [GJ]	μ_{A_a}[m^2]	σ_{A_a}[m^2]	μ_{E_c} [GJ]	σ_{E_c} [GJ]
0.50	19/31	21/31	301.04	230.50	50.78	4.50	1.08	345.49	97.98	7.08	2.36
0.55	22/31	23/31	311.03	241.54	59.24	4.72	1.28	364.59	64.96	7.41	1.42
0.60	27/31	27/31	319.46	236.11	36.57	4.58	0.78	300.96	62.96	6.07	1.35
0.65	29/31	30/31	278.19	229.56	46.42	4.46	1.06	331.52	18.10	6.78	0.15
0.70	26/31	29/31	304.03	219.22	47.12	4.25	1.02	318.46	105.72	6.51	2.32
0.75	28/31	28/31	263.91	214.38	39.63	4.12	0.82	341.55	25.29	6.72	0.59
0.80	29/31	31/31	315.87	208.55	32.82	4.00	0.68	228.48	39.82	4.35	0.83
0.85	29/31	30/31	278.99	208.64	41.20	4.01	0.89	286.72	89.60	5.45	1.71
0.90	30/31	30/31	277.83	200.00	35.22	3.79	0.74	216.96	–	4.20	–
0.95	29/31	31/31	261.50	183.46	29.76	3.47	0.66	328.64	45.71	7.28	1.74

at t =20 s), with one drone being lost every 15 s thereafter. Thus, if the swarm is unsuccessful, 70 drones will survive after fighting the propagation of the fire for 480 s. The experimental results are provided in Table 6 and Fig. 16.

The output data files resulting from the numerical experiments in this section are available in [dataset] [38].

As can be observed in Table 6, the success rate is about 65% with an average swarm-size of 75 at the end of the firefighting operations. This is down from the 90% shown in Table 4 for ω = 0.75 and 100 drones. While performance will always decrease when drone losses occur, this experiment shows that the system adapts to the situation and remains operational. In addition, note that the curves for URs in Fig. 16 do not exhibit an exponential but an approximately linear increase (i.e. the curves are neither concave nor convex). This means that, although the fire is not being fully suppressed in those unsuccessful cases, it is not going out of control either. This is confirmed by observing the animations in [dataset] [38] (e.g. see animations corresponding to Runs 1, 6 and 7).

Fig. 17 shows a successful run despite the loss of 21 drones. In turn, Fig. 18 shows the only run in which the swarm mistakenly deemed the fire suppressed and returned to the docking stations only to allow the missed hotspot to propagate unchecked for the remaining time of the simulation.

6. Conclusions and Future Work

Destructive wildfires continue to increase in frequency and severity worldwide. Modern drones are often used to operate in such dangerous environments for monitoring and detection of wildfires but scarcely for their suppression. This work investigated the feasibility and potential of using self-organising swarms of drones to fight the propagation of wildfires autonomously and collaboratively, without risking human lives.

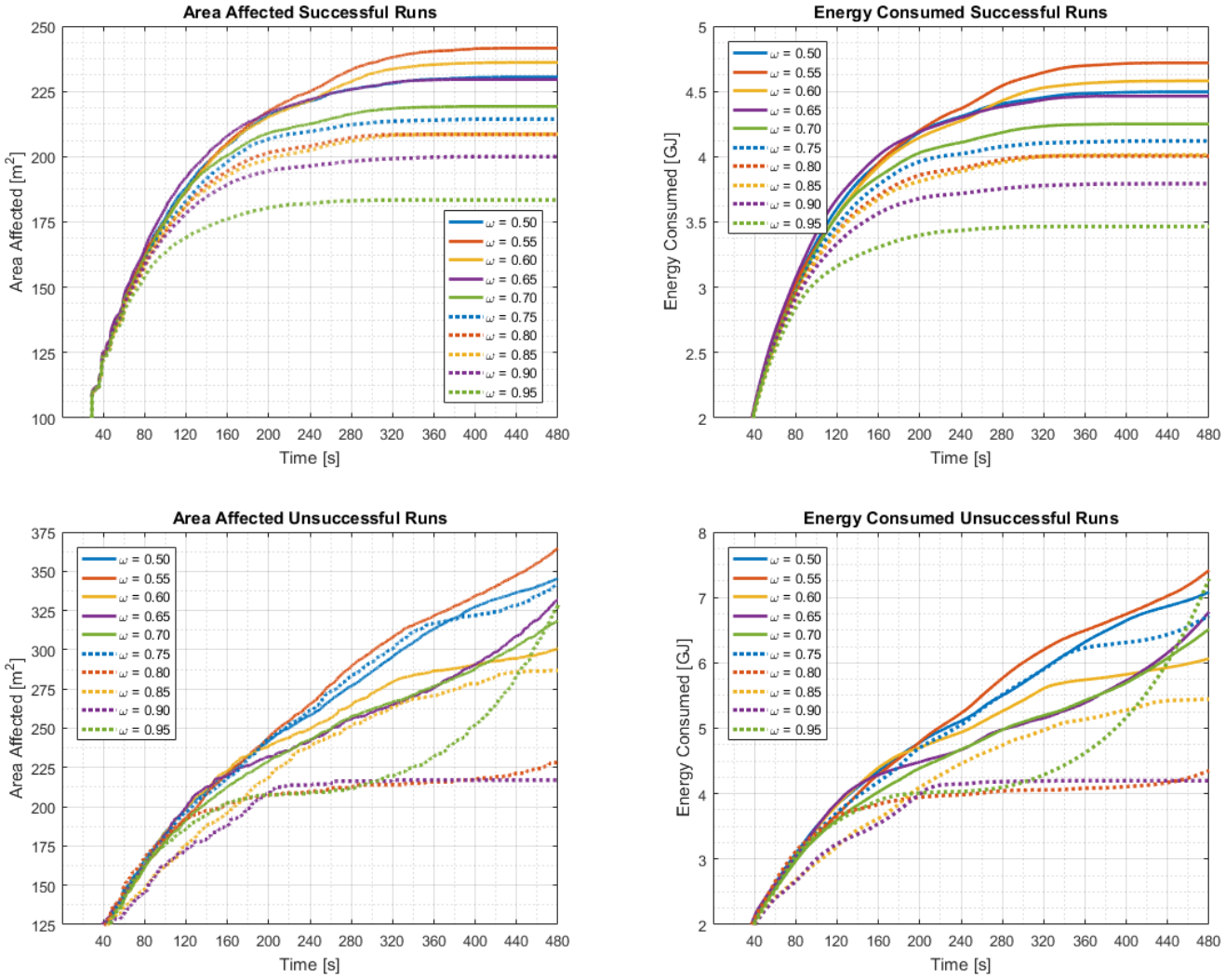


Figure 15: Mean A_a and mean E_c by a fire originated from four sources corresponding to a swarm of 100 drones fighting its propagation for 480 s with different values of the inertia weight (ω) between 0.50 and 0.95. Whilst successful (SRs) and unsuccessful runs (URs) are considered separately, the mean values are calculated out of data gathered from 31 runs in total for each ω. Hence the sum of the numbers of SRs and URs for a given ω equals 31.

The focus was not on the design of the physical robots but on their self-coordination mechanisms so that the desired firefighting behaviour would emerge. Thus, an efficient physics-based model of fire propagation and a self-organisation algorithm for swarms of firefighting drones were developed and coupled, with the collaborative behaviour based on a particle swarm algorithm adapted to individuals operating within physical dynamic environments of high severity and frequency of change.

The numerical experiments carried out in section 5 demonstrated that the calibrated physics-based fire propagation model could make realistic predictions, confirmed the expected scalability and fault-tolerance of the proposed self-organising swarm of drones as an autonomous firefighting system, and provided some insight into potential parameters to control the emergent behaviour of the swarm. It can be concluded that the exploitation of self-organising decentralised collaborative behaviour is a powerful approach and a promising line of research to deal with complex dynamic problems such as the suppression of wildfires. In principle, any wildfire could be suppressed given enough time and a sufficiently high number of firefighting drones – even if this number is unrealistically high with current technology.

It is important to note that, whilst water was used as the fire suppressant, the ultimate goal is to exploit more sophisticated technology with higher extinguishing power and comprising a lighter payload for the drones. One key challenge for this technology to become practical is the design of efficient, lightweight and environmentally friendly fire-suppression materials which can be carried by relatively small drones. One of our future research directions lies in the design of such materials based on Class-A foam enhanced by nanoparticles.

At present, we are working towards the incorporation of mass and momentum balance into the fire model, and of flight dynamics and multi-agent collision-avoidance algorithms into the drone swarm model. In this regard, we are exploring the use of charged particles (commonly used to maintain diversity [8])

Table 6: Information extracted 480 s after a fire was originated from four sources, with an initial swarm of 100 drones (ω =0.75) fighting its propagation. Drone losses occur during firefighting operations at the constant rate of one unit every 15 s, with the first drone lost at t = 30 s. While 31 runs are carried out for the statistics, means (μ) and standard deviations (σ) are calculated separately for successful runs (SRs) and unsuccessful runs (URs). A fire is deemed suppressed when the swarm believes it to be so, and returns to the docking stations. Total area with fuel is 6,529 m^2 and total fuel energy before fire is 218.30 GJ.

Mean	SRs	SRs	Mean	Successful Runs (SRs)				Unsuccessful Runs (URs)			
m_{final}	Actual	Deemed	t_{sup} [s]	μ_{aa}[m^2]	σ_{aa}[m^2]	μ_{ec} [GJ]	σ_{ec} [GJ]	μ_{aa}[m^2]	σ_{aa}[m^2]	μ_{ec} [GJ]	σ_{ec} [GJ]
75.45	20/31	21/31	327.36	217.63	36.54	4.19	0.74	423.56	108.08	8.98	2.52

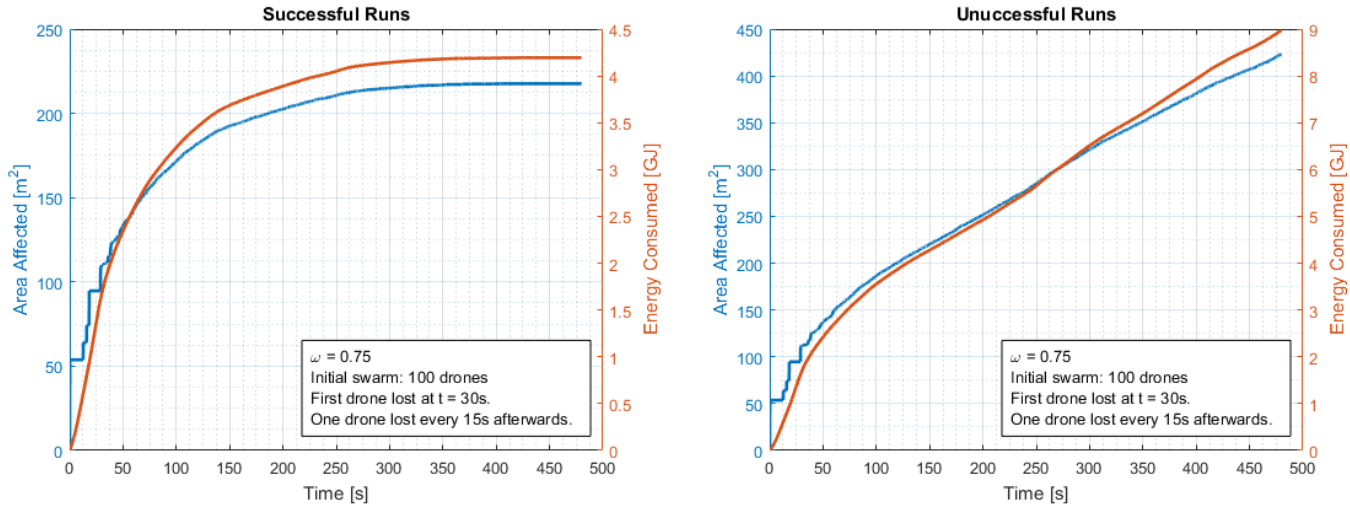


Figure 16: Mean A_{a} and mean E_{c} by a fire originated from four sources corresponding to an initial swarm of 100 drones fighting its propagation for 480 s (ω =0.75). Drone losses occur during firefighting operations at the constant rate of one unit every 15 s, with the first drone lost at t = 30 s. While 31 runs are carried out for the statistics, the mean values are calculated separately for successful and unsuccessful runs.

or potential field algorithms [78, 77]. Other lines of future research include different neighbourhoods (whether topological, distance-based or speciation); mechanisms to handle the *outdated memory* and the *diversity loss*; and the development of self-organisation algorithms to achieve more sophisticated firefighting behaviour than the one that emerges from simply searching for hotspots.

References

[1] Almeida, R.M., Macau, E.E.N., 2011. Stochastic cellular automata model for wildland fire spread dynamics. Journal of Physics: Conference Series 285, 012038. doi:10.1088/1742-6596/285/1/012038.

[2] Amikiya, A.E., Banda, M.K., 2018. Modelling and simulation of reactive transport phenomena. Journal of Computational Science doi:10.1016/j.jocs.2018.08.002.

[3] Anderson, H.E., 1982. Aids to determining fuel models for estimating fire behavior. Technical Report INT-122. USDA Forest Service, Intermountain Forest and Range Experiment Station.

[4] Andrews, P.L., 2014. Current status and future needs of the BehavePlus fire modeling system. International Journal of Wildland Fire 23, 21. doi:10.1071/wf12167.

[5] Andrews, P.L., 2018. The rothermel surface fire spread model and associated developments: A comprehensive explanation. Gen. Tech. Rep. RMRS-GTR-371. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station. 121 p. 371.

[6] Arshinov, V., Fuchs, C., 2003. Causality, emergence, self-organisation. NIA-Priroda Moscow.

[7] Benzi, R., Succi, S., Vergassola, M., 1992. The lattice boltzmann equation: theory and applications. Physics Reports 222, 145–197. doi:10.1016/0370-1573(92)90090-m.

[8] Blackwell, T., Branke, J., 2006. Multiswarms, exclusion, and anti-convergence in dynamic environments. IEEE Transactions on Evolutionary Computation 10, 459–472. doi:10.1109/tevc.2005.857074.

[9] Blackwell, T.M., Bentley, P.J., 2002. Dynamic search with charged swarms, in: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc.. pp. 19–26.

[10] Bonabeau, E., Marco, D.d.R.D.F., Dorigo, M., Théraulaz, G., Theraulaz, G., et al., 1999. Swarm intelligence: from natural to artificial systems. 1, Oxford university press.

[11] Bova, A.S., Mell, W.E., Hoffman, C.M., 2016. A comparison of level set and marker methods for the simulation of wildland fire front propagation. International Journal of Wildland Fire 25, 229. doi:10.1071/wf13178.

[12] Box, G.E., Draper, N.R., 1987. Empirical model-building and response surfaces. John Wiley & Sons.

[13] Bui, L.T., Soliman, O., Abbass, H.A., 2007. A modified strategy for the constriction factor in particle swarm optimization, in: Australian Conference on Artificial Life, Springer. pp. 333–344.

[14] Carlisle, A., Dozier, G., 2000. Adapting particle swarm optimization to dynamic environments, in: International conference on artificial intelligence, pp. 429–434.

[15] Casbeer, D.W., Beard, R.W., McLain, T.W., Li, S.M., Mehra, R.K., 2005. Forest fire monitoring with multiple small uavs, in: American Control Conference, 2005. Proceedings of the 2005, IEEE. pp. 3530–3535.

[16] Chen, S., Liu, Z., Tian, Z., Shi, B., Zheng, C., 2008. A simple lattice boltzmann scheme for combustion simulation. Computers & Mathematics with Applications 55, 1424–1432. doi:10.1016/j.camwa.2007.08.020.

[17] Chen, Y.p., Jiang, P., 2010. Analysis of particle interaction in particle swarm optimization. Theoretical Computer Science 411, 2101–2115.

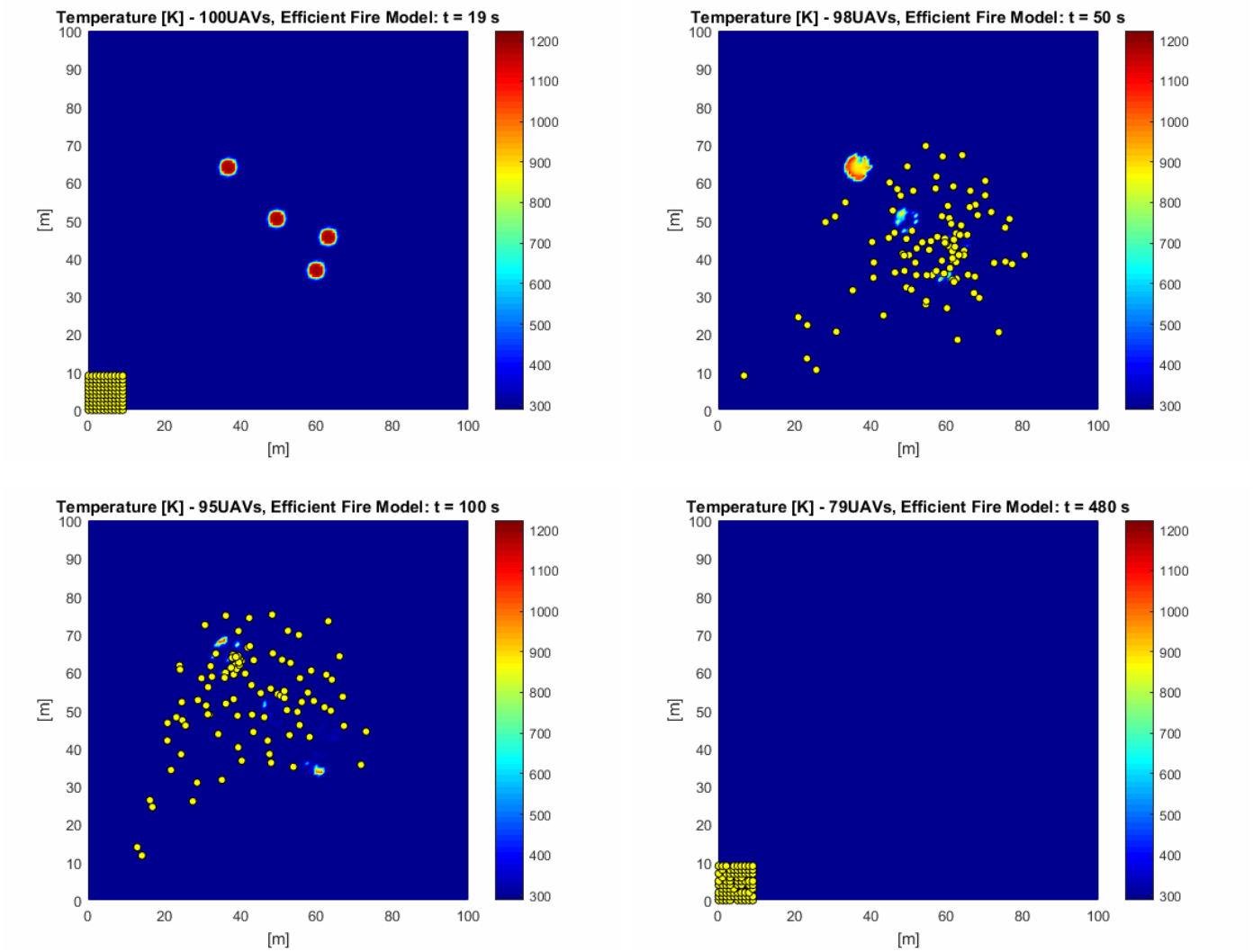


Figure 17: Initial swarm of 100 drones ($\omega = 0.75$) fighting a fire originated from four sources and propagating for 480 s. Drone losses occur during firefighting operations at the constant rate of one unit every 15 s, with the first drone lost at $t = 30$ s. Drones are launched 20 s after the fire is ignited. The figure shows four snapshots of the temperature field at t =19 s, t =50 s, t =100 s and t =480 s. The fire is successfully suppressed, with 79 drones surviving the operations. The animation corresponding to this run is available in [dataset] [38], Setting 1, Run 2.

doi:DOI: 10.1016/j.tcs.2010.03.003. swarm Intelligence Theory: A Snapshot of the State of the Art.

[18] Cheney, N., Gould, J., Catchpole, W., 1998. Prediction of fire spread in grasslands. International Journal of Wildland Fire 8, 1. doi:10.1071/wf9980001.

[19] Chiavazzo, E., Karlin, I.V., Gorban, A.N., Boulouchos, K., 2009. Combustion simulation via lattice boltzmann and reduced chemical kinetics. Journal of Statistical Mechanics: Theory and Experiment 2009, P06013. doi:10.1088/1742-5468/2009/06/p06013.

[20] Chung, S.J., Paranjape, A.A., Dames, P., Shen, S., Kumar, V., 2018. A survey on aerial swarm robotics. IEEE Transactions on Robotics 34, 837–855. doi:10.1109/tro.2018.2857475.

[21] Clerc, M., Kennedy, J., 2002. The particle swarm – explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6, 58–73. doi:10.1109/4235.985692.

[22] Dai D, Z.Y., 2013. Simulating fire spread in a community using an agent-based model, in: Proceedings of the 12th International Conference on GeoComputation. LIESMARS Wuhan University, Wuhan, China, pp. 130–132.

[23] Encinas, L.H., White, S.H., del Rey, A.M., Sánchez, G.R., 2007. Modelling forest fire spread using hexagonal cellular automata. Applied Mathematical Modelling 31, 1213–1227. doi:10.1016/j.apm.2006.04.001.

[24] FCPG022, 2014. Building wildfire resilience into forest management planning: Forestry commission practice guide. Edinburgh, UK: Forestry Commission .

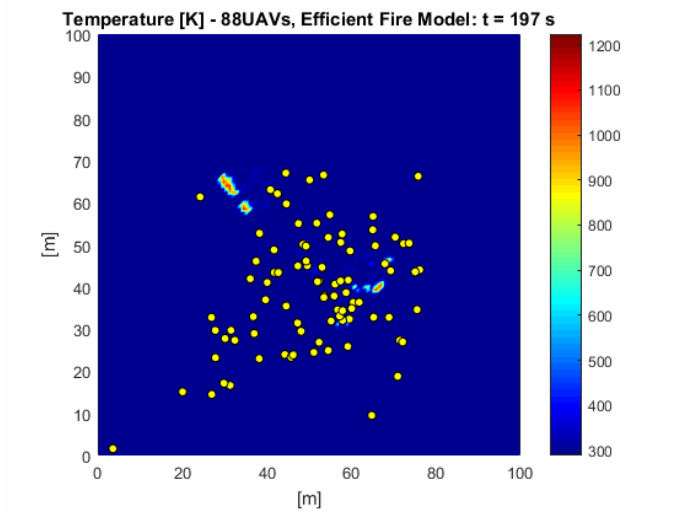
[25] Fernandez-Anez, N., Christensen, K., Rein, G., 2017. Two-dimensional model of smouldering combustion using multi-layer cellular automaton: The role of ignition location and direction of airflow. Fire Safety Journal 91, 243–251. doi:10.1016/j.firesaf.2017.03.009.

[26] Ferragut, L., Asensio, M., Monedero, S., 2007. A numerical method for solving convection–reaction–diffusion multivalued equations in fire spread modelling. Advances in Engineering Software 38, 366–371. doi:10.1016/j.advengsoft.2006.09.007.

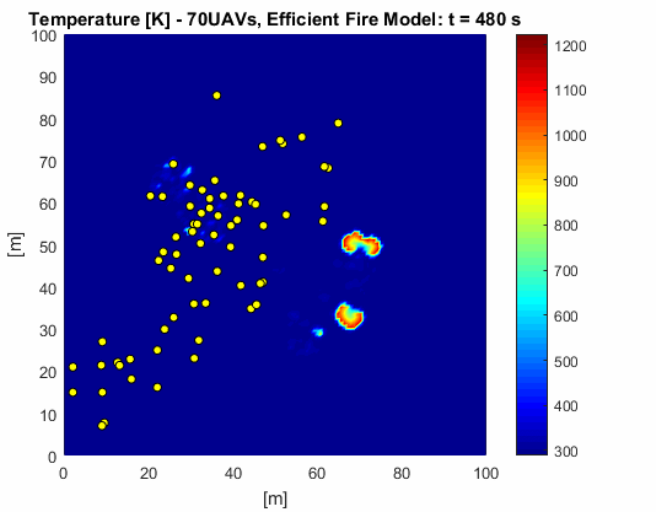
[27] Finney, M.A., 2004. FARSITE: Fire Area Simulator - Model development and evaluation. Technical Report. U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station.

[28] Ghosh, S., Das, S., Kundu, D., Suresh, K., Abraham, A., 2012. Inter-particle communication and search-dynamics of lbest particle swarm optimizers: An analysis. Information Sciences 182, 156–168. doi:10.1016/j.ins.2010.10.015. nature-Inspired Collective Intelligence in Theory and Practice.

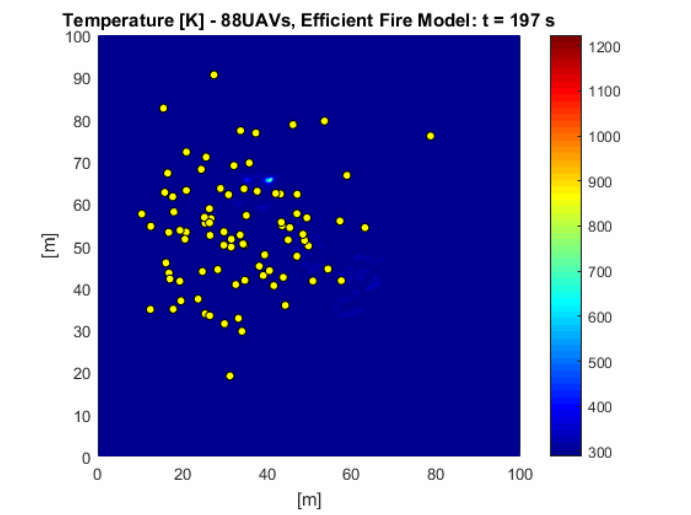
[29] Glasa, J., Halada, L., 2008. On elliptical model for forest fire spread modeling and simulation. Mathematics and Computers in Simulation 78, 76–88. doi:10.1016/j.matcom.2007.06.001.



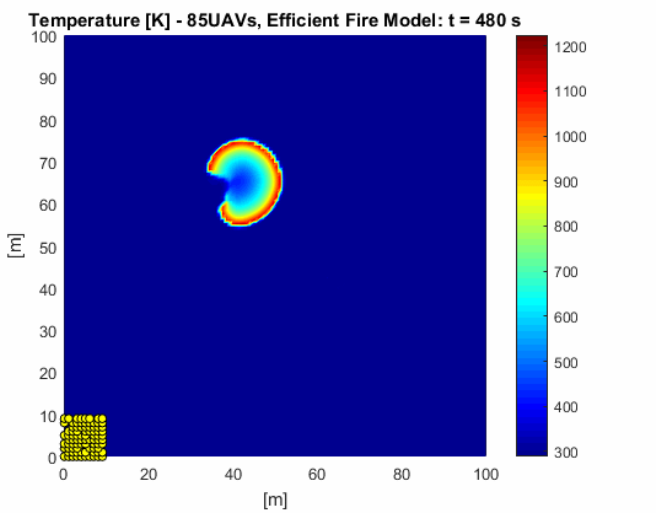
(a) Run 1 of 31, t = 197s, 88 drones left.



(b) Run 1 of 31, t = 480s, 70 drones left.



(c) Run 8 of 31, t = 197s, 88 drones left.



(d) Run 8 of 31, t = 480s, 85 drones left.

Figure 18: Initial swarm of 100 drones (ω = 0.75) fighting a fire originated from four sources and propagating for 480 s. Drone losses occur during firefighting operations at the constant rate of one unit every 15 s, with the first drone lost at t = 30 s. Drones are launched 20 s after the fire is ignited. The figure shows two snapshots of the temperature field at t =197 s and t =480 s for an unsuccessful run in [dataset] [38], Setting 1, Run 1 (top) and for a run that the swarm mistakenly deemed successful in [dataset] [38], Setting 1, Run 8 (bottom).

[30] Graml, R., Wigley, G., 2008. Bushfire hotspot detection through uninhabited aerial vehicles and reconfigurable computing, in: 2008 IEEE Aerospace Conference, IEEE. doi:10.1109/aero.2008.4526475.

[31] Grasso, P., Innocente, M.S., 2018. A two-dimensional reaction-advection-diffusion model of the spread of fire in wildlands. Advances in forest fire research .

[32] Group, C.F.C.F.D., Science, C.F.C., Directorate, S.D., 1992. Development and structure of the Canadian forest fire behavior prediction system. volume 3. Forestry Canada, Science and Sustainable Development Directorate.

[33] Heppner, F., Grenander, U., 1990. A stochastic nonlinear model for coordinated bird flocks. The ubiquity of chaos 223.

[34] Heyligen, F., 2002. The science of self-organisation and adaptivity, in: The Encyclopedia of Life Support Systems. UNESCO Publishing-Eolss Publishers.

[35] Heylighen, F., Gershenson, C., 2003. The meaning of self-organization in computing. IEEE Intelligent Systems 18.

[36] Innocente, M., 2019a. All Figures. Figshare. Fileset. Doi: 10.6084/m9.figshare.7163402.v5.

[37] Innocente, M., 2019b. Control Coefficients Experiments. Figshare. Fileset. Doi: 10.6084/m9.figshare.7150967.v4.

[38] Innocente, M., 2019c. Fault Tolerance Experiments. Figshare. Fileset. Doi: 10.6084/m9.figshare.7550111.

[39] Innocente, M., 2019d. Fire Model Calibration and Fire Propagation Experiments. Figshare. Fileset. Doi: 10.6084/m9.figshare.7147124.v4.

[40] Innocente, M., 2019e. Scalability Experiments. Figshare. Fileset. Doi: 10.6084/m9.figshare.7150898.v3.

[41] Innocente, M.S., Afonso, S.M.B., Sienz, J., Davies, H.M., 2015. Particle swarm algorithm with adaptive constraint handling and integrated surrogate model for the management of petroleum fields. Applied Soft Computing 34, 463–484. doi:10.1016/j.asoc.2015.05.032.

[42] Innocente, M.S., Grasso, P., 2018. Swarms of autonomous drones self-organised to fight the spread of wildfires, in: Proceedings of the GEOSAFE Workshop on Robust Solutions for Fire Fighting.

[43] Innocente, M.S., Sienz, J., 2010. Coefficients' settings in particle swarm optimization: Insight and guidelines, in: Dvorkin, E., Goldschmit, M., Storti, M. (Eds.), Mecom-Cilamce 2010, Asociación Argentina de Mecánica Computacional. pp. 9253–9269.

[44] Innocente, M.S., Sienz, J., 2011. Particle swarm optimization with inertia weight and constriction factor, in: Proceedings of the 2011 International conference on swarm intelligence (ICSI 2011), pp. id–1–id–11.

[45] Kacem, A., Lallemand, C., Giraud, N., Mense, M., Gennaro, M.D., Pizzo, Y., Loraud, J.C., Boulet, P., Porterie, B., 2017. A small-world network model for the simulation of fire spread onboard naval vessels. Fire Safety Journal 91, 441–450. doi:10.1016/j.firesaf.2017.04.009.

[46] Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks, pp. 1942–1948. doi:10.1109/ICNN.1995.488968.

[47] Kennedy, J., Eberhart, R.C., Shi, Y., 2001. Swarm intelligence. The Morgan Kaufmann series in evolutionary computation, Morgan Kaufmann Publishers.

[48] Leonard, B.J., Engelbrecht, A.P., 2012. Scalability study of particle swarm optimizers in dynamic environments, in: Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 121–132. doi:10.1007/978-3-642-32650-9_11.

[49] Li, C., Yang, S., 2013. A comparative study on particle swarm optimization in dynamic environments, in: Studies in Computational Intelligence. Springer Berlin Heidelberg, pp. 109–136. doi:10.1007/978-3-642-38416-5_5.

[50] Li, X., 2004. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization, in: Genetic and Evolutionary Computation Conference, Springer. pp. 105–116.

[51] Li, X., Branke, J., Blackwell, T., 2006. Particle swarm with speciation and adaptation in a dynamic environment, in: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM. pp. 51–58.

[52] Linn, R., Reisner, J., Colman, J.J., Winterkamp, J., 2002. Studying wildfire behavior using FIRETEC. International Journal of Wildland Fire 11, 233. doi:10.1071/wf02007.

[53] Lynn, N., Ali, M.Z., Suganthan, P.N., 2018. Population topologies for particle swarm optimization and differential evolution. Swarm and Evolutionary Computation 39, 24 – 35. doi:https://doi.org/10.1016/j.swevo.2017.11.002.

[54] Mallet, V., Keyes, D., Fendell, F., 2009. Modeling wildland fire propagation with level set methods. Computers & Mathematics with Applications 57, 1089–1101. doi:10.1016/j.camwa.2008.10.089.

[55] Margerit, J., Séro-Guillaume, O., 2002. Modelling forest fires. part II: reduction to two-dimensional models and simulation of propagation. International Journal of Heat and Mass Transfer 45, 1723–1737. doi:10.1016/s0017-9310(01)00249-6.

[56] McGrattan, K., Hostikka, S., McDermott, R., Floyd, J., Weinschenk, C., Overholt, K., 2013. Fire Dynamics Simulator Technical Reference Guide Volume 1: Mathematical Model [6th Ed.]. Technical Report. National Institute of Standards and Technology.

[57] McGrattan, K., Hostikka, S., McDermott, R., Floyd, J., Weinschenk, C., Overholt, K., 2017. Fire Dynamics Simulator Technical Reference Guide Volume 3: Validation [6th Ed.]. Technical Report. National Institute of Standards and Technology.

[58] Mell, W., Jenkins, M.A., Gould, J., Cheney, P., 2007. A physics-based approach to modelling grassland fires. International Journal of Wildland Fire 16, 1. doi:10.1071/wf06002.

[59] Mitchell, M., 2009. Complexity: A guided tour. Oxford University Press.

[60] Ntinas, V.G., Moutafis, B.E., Trunfio, G.A., Sirakoulis, G.C., 2017. Parallel fuzzy cellular automata for data-driven simulation of wildfire spreading. Journal of Computational Science 21, 469–485. doi:10.1016/j.jocs.2016.08.003.

[61] Pak, S.I., Hayakawa, T., 2011. Forest fire modeling using cellular automata and percolation threshold analysis, in: Proceedings of the 2011 American Control Conference, IEEE. doi:10.1109/acc.2011.5991603.

[62] Parrott, D., Li, X., 2006. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. IEEE Transactions on Evolutionary Computation 10, 440–458. doi:10.1109/tevc.2005.859468.

[63] Parvin, H., Minaei, B., Ghatei, S., 2011. A new particle swarm optimization for dynamic environments, in: Computational Intelligence in Security for Information Systems. Springer Berlin Heidelberg, pp. 293–300. doi:10.1007/978-3-642-21323-6_37.

[64] Perry, G., 1998. Current approaches to modelling the spread of wildland fire: a review. Progress in Physical Geography 22, 222–245.

[65] Porterie, B., Zekri, N., Clerc, J.P., Loraud, J.C., 2007. Modeling forest fire spread and spotting process with small world networks. Combustion and Flame 149, 63–78. doi:10.1016/j.combustflame.2006.12.008.

[66] Reynolds, C.W., 1987. Flocks, herds and schools: A distributed behavioral model, in: ACM SIGGRAPH computer graphics, ACM. pp. 25–34.

[67] Rezazadeh, I., Meybodi, M.R., Naebi, A., 2011. Adaptive particle swarm optimization algorithm in dynamic environments, in: Third International Conference on Computational Intelligence, Modelling & Simulation, IEEE. doi:10.1109/cimsim.2011.23.

[68] Richards, G.D., 1990. An elliptical growth model of forest fire fronts and its numerical solution. International Journal for Numerical Methods in Engineering 30, 1163–1179.

[69] Rios, O., Jahn, W., Rein, G., 2014. Forecasting wind-driven wildfires using an inverse modelling approach. Natural Hazards and Earth System Science 14, 1491–1503. doi:10.5194/nhess-14-1491-2014.

[70] Rothermel, R.C., et al., 1972. A mathematical model for predicting fire spread in wildland fuels .

[71] Rui, X., Hui, S., Yu, X., Zhang, G., Wu, B., 2017. Forest fire spread simulation algorithm based on cellular automata. Natural Hazards 91, 309–319. doi:10.1007/s11069-017-3127-5.

[72] Sadeghi, S., Parvin, H., Rad, F., 2015. Particle swarm optimization algorithm for dynamic environments, in: Lecture Notes in Computer Science. Springer International Publishing, pp. 260–269. doi:10.1007/978-3-319-27060-9_21.

[73] Şahin, E., 2005. Swarm robotics: From sources of inspiration to domains of application, in: Swarm Robotics. Springer Berlin Heidelberg, pp. 10–20. doi:10.1007/978-3-540-30552-1_2.

[74] Şahin, E., Girgin, S., Bayindir, L., Turgut, A.E., 2008. Swarm robotics, in: Swarm Intelligence: Introduction and Applications. Springer-Verlag Berlin Heidelberg, pp. 87–100.

[75] Séro-Guillaume, O., Margerit, J., 2002. Modelling forest fires. part i: a complete set of equations derived by extended irreversible thermodynamics. International Journal of Heat and Mass Transfer 45, 1705–1722. doi:10.1016/s0017-9310(01)00248-4.

[76] Shi, Y., Eberhart, R.C., 1998. A modified particle swarm optimizer, in: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (CEC 1998), pp. 69–73. doi:10.1109/ICEC.1998.699146.

[77] Statheros, T., Howells, G., Lorrentz, P., McDonald-Maier, K., 2009. A novel potential field algorithm and an intelligent multi-classifier for the automated control and guidance system, pp. 337–342.

[78] Statheros, T., Howells, G., McDonald-Maier, K., 2007. Trajectory equilibrium state detection and avoidance algorithm for multi-autonomous potential field mobile robots. Electronics Letters 43, 799. doi:10.1049/el:20071197.

[79] Succi, S., Benzi, R., Higuera, F., 1991. The lattice boltzmann equation: A new tool for computational fluid-dynamics. Physica D: Nonlinear Phenomena 47, 219–230. doi:10.1016/0167-2789(91)90292-h.

[80] Sullivan, A.L., a. A review of wildland fire spread modelling, 1990-present, 1: Physical and quasi-physical models doi:10.1071/WF06143, `arXiv:0706.3074v1`.

[81] Sullivan, A.L., b. A review of wildland fire spread modelling, 1990-present 2: Empirical and quasi-empirical models doi:10.1071/WF06142, `arXiv:0706.4128v1`.

[82] Sullivan, A.L., c. A review of wildland fire spread modelling, 1990-present 3: Mathematical analogues and simulation models doi:10.1071/WF06144, `arXiv:0706.4130v1`.

[83] Tolhurst, K., Shields, B., Chong, D., et al., 2008. Phoenix: development and application of a bushfire risk management tool. Australian Journal of Emergency Management, The 23, 47.

[84] Torresan, C., Berton, A., Carotenuto, F., Di Gennaro, S.F., Gioli, B., Matese, A., Miglietta, F., Vagnoli, C., Zaldei, A., Wallace, L., 2017. Forestry applications of uavs in europe: A review. International Journal of Remote Sensing 38, 2427–2447.

[85] Tymstra, C., Bryce, R., Wotton, B., Taylor, S., Armitage, O., et al., 2010. Development and structure of prometheus: the canadian wildland fire growth simulation model. Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre, Information Report NOR-X-417.(Edmonton, AB) .

[86] Xia, X., Gui, L., Zhan, Z.H., 2018. A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting. Applied Soft Computing 67, 126 – 140. doi:https://doi.org/10.1016/j.asoc.2018.02.042.

[87] Yamamoto, K., Takada, N., Misawa, M., 2005. Combustion simulation with lattice boltzmann method in a three-dimensional porous structure. Proceedings of the Combustion Institute 30, 1509–1515. doi:10.1016/j.proci.2004.08.030.

[88] Yazdani, D., Nasiri, B., Sepas-Moghaddam, A., Meybodi, M.R., 2013. A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization. Applied Soft Computing 13, 2144–2158. doi:10.1016/j.asoc.2012.12.020.

[89] Yuan, C., Zhang, Y., Liu, Z., 2015. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. Canadian journal of forest research 45, 783–792.

[90] Zhang, L., Yu, H., Hu, S., 2003. A new approach to improve particle swarm optimization, in: Genetic and Evolutionary Computation Conference, Springer. pp. 134–139.