

Swarms of autonomous drones self-organised to fight the spread of wildfires

Mauro S. Innocente
Smart Vehicles Control Laboratory
(SVeCLab), Coventry University
Coventry, CV1 2TL
Mauro.S.Innocente@coventry.ac.uk

Paolo Grasso
Smart Vehicles Control Laboratory
(SVeCLab), Coventry University
Coventry, CV1 2TL
grassop@uni.coventry.ac.uk

Abstract

Swarm robotics and drone technology have progressed at an increasingly fast pace for the past two decades, extending their capabilities and the kinds of problems they can help tackle. Drones can now be equipped with a range of advanced devices and sensors which enable them to navigate remote areas and to operate in dangerous environments. Given the hazardous nature of the activity, fighting fires by means of disposable and relatively inexpensive robots in place of humans is of special interest. In addition, the use of fleets of decentralised cooperative and self-organising robots results in a robust and resilient system with collective decision-making which can cope with uncertainty, errors, and the failure or loss of a few non-essential units without jeopardising the mission. This paper comprises an initial proof of concept to demonstrate the feasibility and potential of employing swarm robotics to fight fires autonomously. Thus, an efficient yet realistic model of the spread of wildfires is developed, which is then coupled with a model of a fleet of self-organising drones whose coordination mechanism is based on a forgetful particle swarm algorithm.

1 Introduction

Swarm Intelligence (SI) comprises a relatively novel route to Artificial Intelligence (AI) which stems from decentralised and self-organising behaviour observed in groups of simple social animals in nature such as ant colonies, bee colonies, fish schools, and bird flocks. By way of collaboration, these animals are able to accomplish tasks that are far beyond their individual capabilities, and even beyond the aggregation of all their individual capabilities. That is to say, 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. Swarm robotics is a novel approach to the coordination of large numbers of simple and relatively inexpensive robots, which emerged as the application of SI to multi-robot systems. Different from other SI studies, swarm robotics puts emphasis on the physical embodiment of individuals [Sah05].

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: G. Di Stefano, A. Navarra Editors: Proceedings of the RSFF'18 Workshop, L'Aquila, Italy, 19-20-July-2018, published at <http://ceur-ws.org>

Swarm robotics and drone technology have progressed at an increasingly fast pace for the past two decades, extending their capabilities and the kinds of problems they can help tackle. Drones can now be equipped with a range of advanced cameras, thermal imagers and sensors which enable them to operate in remote areas, dangerous environments, and even through solid smoke while still being able to perform tasks such as surveying, mapping or locating people. Some of their current applications include aerial photography and filming, information gathering for human decision-makers, provision of essential supplies for disaster management, support of search and rescue operations, mapping inaccessible locations, field surveying, and crop health monitoring. With regards to fire-fighting, drone technology has been applied to forest surveillance, building fire risk maps, forest fire detection and monitoring [CBM⁺05], post-fire recovery monitoring and forest fire measuring [TBC⁺17], assisting search and rescue operations, and bushfire hotspot detection [GW08]. A survey on technologies for automatic forest fire monitoring, detection, and fighting using drones can be found in [YZL15].

Robotics and Autonomous Systems (RAS) are predicted to play an increasingly important role in disaster relief by reducing costs and response times and increasing capabilities [UK-17]. Given the hazardous nature of the activity, fighting fires by means of disposable and relatively inexpensive robots in place of humans is of special interest. In addition, the use of fleets of decentralised cooperative and self-organising robots results in a robust and resilient system with collective decision-making which can cope with uncertainty, errors, and the failure or loss of a few non-essential units without jeopardising the overall mission. However, the use of self-organising swarms of autonomous drones for the actual deed of putting out fires has remained notably unexplored.

This paper comprises a proof of concept to demonstrate the feasibility and potential of employing swarm robotics to fight fires autonomously. To this end, an efficient yet realistic physics-based model of the spread of wildfires is developed, which is coupled with a model of a fleet of self-organising drones whose coordination mechanism is based on a forgetful particle swarm algorithm. The aim is to develop algorithms for swarms of drones to self-organise so as to develop the ability to fight the spread of wildfires without human intervention.

The remainder of the paper is organised as follows: section 2 presents a short introduction to the modelling of the spread of wildfires; section 3 offers an introduction to Particle Swarm Optimisation (PSO), including adaptations to handle dynamic environments; section 4 presents the proposed model of self-organising fire-fighting drones, where an efficient physics-based model of the spread of fire is introduced in section 4.1 and a PSO-based coordination mechanism is developed in section 4.2 for the self-organisation of the swarm; section 5 presents initial simulations; whilst concluding remarks and future work are proposed in section 6.

2 Modelling of the Spread of Fire

Depending on the features that are taken into account, different taxonomies are possible to classify models of the spread of fire. Possibly the most general one is to classify them as either *phenomenological* (black-box) or *physics-based* (grey and white box) models.

Cellular Automata (CA) 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 the states of the neighbouring cells. When used as emulators, they are able to represent complex physics with a very high level of efficiency and robustness. Within the fire-spread modelling domain, CA are used as phenomenological models –i.e. as emulators, not derived from first principles. Thus, they are designed independently from the underlying theories of fire dynamics but based on a few leading assumptions. Prominent examples are the Fire Area Simulator (*FARSITE*) [Fin04], the Canadian wildland fire growth simulator *Prometheus* [TBW⁺10], and the Australian Bushfire Risk Management Tool *Phoenix* [TSC⁺08].

Loosely speaking, there are two approaches that can be followed when deriving a physics-based model [SGM02]: 1) Bottom-up approach, and 2) Top-down approach. The former starts from rudimentary models with numerous simplifying assumptions, then progressively adding complexity by incorporating previously disregarded phenomena. These are usually semi-empirical, based on simple balance laws, and require empirical parameters and a fair amount of experimental data to calibrate them for some particular environmental condition, which are difficult to obtain and not scalable. Examples are the classical Rothermel’s model [R⁺72] and Rio et al.’s model [RJR14] combining Rothermel’s and Richard’s models [Ric90] into a so-called forward model, and an inverse modelling approach that minimises the differences between the predictions of the forward model and sensor observations. Conversely, the top-down approach consists of formulating the model from first principles in its most general form, including as many of the relevant physical phenomena as possible, then progressively simplifying the model

as required in order to cope with practical problems such as limited available resources as they arise. They typically solve convection-reaction-diffusion equations, with model parameters aimed to be mathematically derived. Prominent examples are [SGM02, MSG02, FAM07, MHM⁺13].

3 Particle Swarm Optimisation

The Particle Swarm Optimisation (PSO) method was inspired by earlier bird-flock simulations framed within the field of social psychology. 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 [KES01]. 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 weighted average of an individual attractor (particle's best experience) and a social attractor (best experience among those 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, the individual behaviour is controlled by the settings of the coefficients in the trajectory equation.

3.1 Trajectory Difference Equation

The *original PSO* (*OPSO*) algorithm proposed in [KE95] is rearranged into a single equation in (1):

$$x_{ij}^{(t)} = x_{ij}^{(t-1)} + \left(x_{ij}^{(t-1)} - x_{ij}^{(t-2)}\right) + iw_{ij}^{(t)}U_{(0,1)}\left(x_{ij}^{(t-1)} - x_{ij}^{(t-1)}\right) + sw_{ij}^{(t)}U_{(0,1)}\left(x_{kj}^{(t-1)} - x_{ij}^{(t-1)}\right) \quad (1)$$

where $x_{ij}^{(t)}$ is the coordinate j of the *position* of particle i at time-step t ; $x_{kj}^{(t)}$ 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 ; $iw_{ij}^{(t)}$ and $sw_{ij}^{(t)}$ are the *individuality weight* and *sociality weight*, respectively, both originally equal to 2 $\forall i, j, t$; and $U_{(0,1)}$ is a random number from a uniform distribution in the range $[0,1]$, resampled anew every time it is referenced.

Eq. (1) is unstable, and particles tend to diverge from the 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. This restriction helps prevent the explosion but does not ensure convergence or a fine-grain search.

Clerc et al. [CK02] analysed the trajectory of a deterministic particle in *OPSO*, becoming one of the first PSO theorists in developing coefficients (constriction factor, χ) which ensure convergence. The formulation of the trajectory difference equation in Eqs. (2) and (3) is referred to here as the Constricted PSO (CoPSO).

$$x_{ij}^{(t)} = x_{ij}^{(t-1)} + \chi_{ij}^{(t)} \left[\left(x_{ij}^{(t-1)} - x_{ij}^{(t-2)}\right) + iw_{ij}^{(t)}U_{(0,1)}\left(x_{ij}^{(t-1)} - x_{ij}^{(t-1)}\right) + sw_{ij}^{(t)}U_{(0,1)}\left(x_{kj}^{(t-1)} - x_{ij}^{(t-1)}\right) \right] \quad (2)$$

$$\chi_{ij}^{(t)} = \begin{cases} \frac{2\kappa_{ij}^{(t)}}{\left(aw_{ij}^{(t)}\right) - 2 + \sqrt{\left(aw_{ij}^{(t)}\right)^2 - 4\left(aw_{ij}^{(t)}\right)}} & \text{if } aw_{ij}^{(t)} \geq 4 \\ \kappa_{ij}^{(t)} & \text{if } aw_{ij}^{(t)} \in (0, 4) \end{cases} \quad (3)$$

where $aw_{ij}^{(t)} = iw_{ij}^{(t)} + sw_{ij}^{(t)}$ and $\kappa_{ij}^{(t)} \in (0, 1) \forall i, j, t$.

Similarly, the formulation here referred to as *classical PSO* (CPSO) incorporates the inertia weight (ω) initially proposed by Shi et al. [SE98], which can be set so as to also ensure convergence and favour fine-grain search:

$$x_{ij}^{(t)} = x_{ij}^{(t-1)} + \omega_{ij}^{(t)}\left(x_{ij}^{(t-1)} - x_{ij}^{(t-2)}\right) + iw_{ij}^{(t)}U_{(0,1)}\left(x_{ij}^{(t-1)} - x_{ij}^{(t-1)}\right) + sw_{ij}^{(t)}U_{(0,1)}\left(x_{kj}^{(t-1)} - x_{ij}^{(t-1)}\right) \quad (4)$$

where $\omega_{ij}^{(t)}$, $iw_{ij}^{(t)}$, $sw_{ij}^{(t)}$ are the j^{th} components of the *inertia weight*, *individuality weight* and *sociality weight*, respectively, of particle i at time-step t . For inertia weight combined with constriction factor, see [IS11].

3.2 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 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. 1.

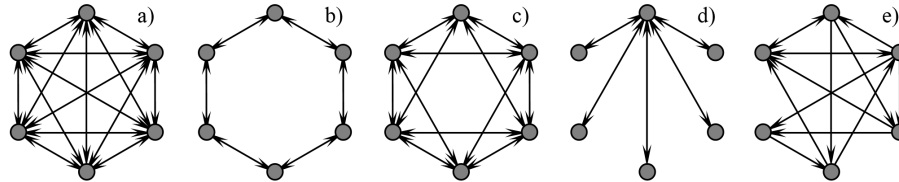


Figure 1: a) global topology; b) ring topology with two neighbours; c) ring topology with four neighbours; d) wheel topology (global for one particle and two neighbours for the rest); e) random topology.

An alternative to traditional topological neighbourhoods is to define the nearest neighbours based on actual Euclidean distance, or using the so-called *speciation* [Li04, LBB06, PL06] where the local attractor (i.e. neighbourhood best) is chosen as the best within adaptively generated species (clusters). For further studies on neighbourhood structures and inter-particle communication, refer to [GDK⁺12, CJ10, LAS18, XGZ18].

3.3 Dynamic Environments

During the execution of a PSO run, if the optimum locations change, the original algorithm has no means to detect this change as particles are still under the influence of their outdated memories. Carlisle et al. [CD00] 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. [BB02] 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 it to better respond to such changes. The drawback is that this method requires the calculation of repulsive forces among all charged particles (n), which presents complexity of $\mathcal{O}(n^2)$. Blackwell et al. [BB06] 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 quantum particles.

Parrot et al. [PL06] 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. [YNSMM13] 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. [SPR15] 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 [LY13, YNSMM13], including diversity-maintaining schemes, memory-based schemes, and multi-swarm schemes. For further reading, refer to [RMN11, PMG11, LE12].

4 Self-Organising Swarm of Fire-Fighting Drones

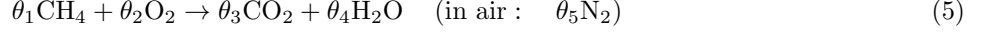
In order to demonstrate the feasibility and potential of employing swarms of drones to fight fires autonomously, without human intervention, realistic models of the spread of wildfires and of the self-organising fire-fighting drones need to be developed and coupled for initial simulations.

4.1 Two-Dimensional Model of the Spread of Wildfires

An efficient yet realistic physics-based fire-spread model was developed, 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 [FAM07], 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 (no transport). In order to compensate for this, the diffusion coefficient is augmented, and two pseudo-3D terms are added into 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 the 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. (5) 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:



The fire-spread model is represented here by a system of five partial differential equations, one for the enthalpy balance and four for the chemical species formation (CO_2 and H_2O) or consumption (Fuel and O_2):

$$\left\{ \begin{array}{l} \rho c_p \frac{\partial T}{\partial t} = \overbrace{-\rho h_c \frac{M}{M_{fuel}} r}^{\text{Combustion}} + \overbrace{k \frac{\partial}{\partial x} \left(\frac{1}{c_p} \frac{\partial (c_p T)}{\partial x} \right) + k \frac{\partial}{\partial y} \left(\frac{1}{c_p} \frac{\partial (c_p T)}{\partial y} \right)}^{\text{Thermal Diffusion}} \\ \quad + \overbrace{k \frac{\partial}{\partial x} \left(\frac{1}{c_p} \frac{\partial h_c}{\partial x} \right) + k \frac{\partial}{\partial y} \left(\frac{1}{c_p} \frac{\partial h_c}{\partial y} \right)}^{\text{Enthalpy Diffusion}} + \overbrace{C_a (T_{\text{amb}} - T)}^{\text{Vertical Convection}} \\ \quad + \sigma \epsilon \left[\overbrace{4 \frac{\partial}{\partial x} \left(T^3 \frac{\partial T}{\partial x} \right) dx + 4 \frac{\partial}{\partial y} \left(T^3 \frac{\partial T}{\partial y} \right) dy}^{\text{2D Radiation}} \right] + \overbrace{\sigma \epsilon \left[\frac{T_{\text{amb}}^4 - T^4}{dz} \right]}^{\text{Vertical Radiation}} \\ \frac{\partial X_i}{\partial t} = -\frac{\theta_i}{\theta_{fuel}} \frac{M}{M_{fuel}} r \quad ; \quad \text{with } i = 1, \dots, 4 \end{array} \right. \quad (6)$$

The system is closed with Eq. (7) for the molar mass of the mixture, Eq. (8) for the heat capacity of the mixture, Eq. (9) for the combustion rate (Arrhenius equation), and Eq. (10) for the combustion enthalpy.

$$M = \sum_{i=1}^5 X_i M_i \quad (7)$$

$$c_p = \sum_{i=1}^5 X_i \frac{M_i}{M} c_{pi} \quad (8)$$

$$r(T, X_1, X_2) = -\delta_{(T, X_1, 2)}^+ A_r T X_1^{0.5} X_2 \exp\left(-\frac{T_a}{T}\right) \quad (9)$$

$$h_c = \frac{H_c(T)}{M} = -\frac{1}{M} \sum_{i=1}^5 \theta_i H_i(T) = \frac{1}{M} \sum_{i=1}^5 \theta_i (H_{i,\text{ref}} + M_i c_{pi} (T_{\text{ref}} - T)) \quad (10)$$

The molar mass of the mixture (M) in Eq. (7) 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. (8) is obtained by weighted summation of the partial heat capacities (c_{pi}).

The combustion rate (r) in Eq. (9) 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). Note that the pre-exponential coefficient (A_r) and the activation temperature (T_a) are empirical parameters.

The function $\delta_{(T, X_{1,2})}^+$ in Eq. (9) is a Kronecker delta as in Eq. (11), which 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.

$$\delta_{(T, X_{1,2})}^+ = \begin{cases} 1 & \text{if } T > T_p \text{ or } X_{1,2} > X_{1,2e} \\ 0 & \text{else} \end{cases} \quad (11)$$

The combustion enthalpy in Eq. (10) 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 found in tables.

4.2 Self-Organisation Model

Since the PSO algorithm was originally inspired by models of decentralised flocks of birds, its potential to prescribe the desired behaviour of a swarm of self-organising drones is very promising. A detailed study of PSO is beyond the scope of this paper. It suffices to say that the attractors in Eq. (4) can be combined into a single attractor at a given time-step (t) as follows [IS10, IASD15]:

$$\phi_{ij}^{(t)} = \iota_{ij}^{(t)} + \sigma_{ij}^{(t)} = i w_{ij}^{(t)} \cdot U_{(0,1)} + s w_{ij}^{(t)} \cdot U_{(0,1)} \quad (12)$$

$$p_{ij}^{(t)} = \frac{\iota_{ij}^{(t)} x b_{ij}^{(t)} + \sigma_{ij}^{(t)} x b_{kj}^{(t)}}{\phi_{ij}^{(t)}} \quad (13)$$

Thus, in classical PSO, the randomness incorporated into the coefficients in the trajectory difference equation affects both the trajectory of a particle towards the unique attractor (p) on the one hand, and the generation of the unique attractor (p) as a stochastic convex combination of the individual and social attractors on the other. Different from classical formulations, these two features of the algorithm are decoupled here. The attractor is generated at every time-step from a uniform distribution within the smallest rectangle that contains the current individual and social attractors. Once the attractor is generated, the trajectory difference equation is as follows:

$$x_{ij}^{(t)} = x_{ij}^{(t-1)} + \omega_{ij}^{(t)} \left(x_{ij}^{(t-1)} - x_{ij}^{(t-2)} \right) + \phi_{ij}^{(t)} \left(p_{ij}^{(t-1)} - x_{ij}^{(t-1)} \right) \quad (14)$$

The random variable ϕ can be realised as in Eq. (12) so that the probability distribution is triangular or trapezoidal (as originally), 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. For optimisation, convergent high-frequency harmonic oscillations are generally preferred (no cost for large displacements). For swarm robotics, however, low-frequency harmonic oscillations and smaller displacements are favoured for obvious reasons. Here we adopt the settings in Eq. (15), which ensure convergent low frequency harmonic oscillations, with ϕ randomly generated within this range from a uniform distribution.

$$\begin{cases} \omega \in (0, 1) \\ \phi \in [(\sqrt{\omega} - 1)^2, (\omega + 1)] \end{cases} \quad (15)$$

While local neighbourhood topologies and other types of sociometries like distance-based nearest neighbours and speciation need to be investigated, only the global neighbourhood topology has been tested here.

As discussed in section 3.3, classical PSO requires some adaptations to be able to cope with dynamic environments. The spread of a wildfire carries all difficulties of a changing environment, even more so if it is being fought: 1) multiple hotspots continually and severely change location, 2) hotspots continually change intensity, 3) hotspots continually appear and vanish throughout the domain, and 4) drones modify the environment (stigmery). Furthermore, re-evaluating particles' memories for the current environment is not realistic, as the sensors of physical drones will not be able to acquire data at distant locations. This issue is dealt with here by resetting a given particle's memory if it has not been updated in the last 15 seconds. If the particle belongs to the first half of the swarm, the reset consists of assigning a random location to the memory with a fictitious temperature equal to the ambient temperature plus 10 degrees. If the particle belongs to the second half of the swarm instead, the memory is simply replaced by the currently held information (both location and temperature).

The search strategy is simple in this first attempt: particles search for the hotspots, and they drop 40 kg of water as soon as their best experience is improved. After three drops, the drone must go back to the water source

to replenish. Although we use water to extinguish the fire in this paper, the ultimate goal is to exploit more sophisticated technology with higher extinguishing power and lower payload. Autonomy of each drone is simply measured in terms of distance travelled since re-filling the fuel tank. As soon as the fuel is fully consumed, the drone must go back to the fuel source to refill the tank.

At this early stage of the research, the swarm of drones has been modelled as 2D massless particles. We are currently working on incorporating collision avoidance algorithms and flight dynamics in the model of the self-organising drones, and transport –including atmospheric wind– in the model of the spread of wildfires.

5 Numerical experiments

The domain used in the experiments is a field of $100 \times 100 \text{ m}^2$ with fuel uniformly distributed everywhere except for a small band around it so that combustion does not take place near the boundaries. The remaining settings are as shown in Table 1:

Table 1: Settings for numerical experiments

R	Universal gas constant	8.3140	$\text{J}\cdot\text{mol}^{-1}\cdot\text{K}^{-1}$
T_{amb}	Ambient temperature	298.1500	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_{ign}	Ignition temperature	573	K
ρ	Gas mixture density	1.2172	$\text{kg}\cdot\text{m}^{-3}$
A_r	Pre-exponential coefficient (Arrhenius)	4.1400×10^{-5}	
C_a	Turbulent convection coefficient in atmosphere	0.0600	$\text{J}\cdot\text{m}^{-3}\cdot\text{K}^{-1}\cdot\text{s}^{-1}$
k	Thermal conductivity (thermal diffusion coefficient)	0.1000	$\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$
c_p	Specific heat capacity at constant pressure	calculated	$\text{J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$
X_f	Initial fuel molar fraction	0.1000	
σ	Stefan-Boltzmann constant	5.6704×10^{-8}	$\text{W}\cdot\text{m}^{-2}\cdot\text{K}^{-4}$
ε	Emissivity factory (< 1)	0.5500	
δx	Optical absorption length (must be smaller than cell size)	0.1000	m
h_c	Specific combustion enthalpy	calculated	$\text{J}\cdot\text{kg}^{-1}$
Δt drone	Time between sensor measurements and memory updates	1	s
Δt fire	Time between fire-spread updates	0.2500	s
$\Delta x = \Delta y$	Cell-size	≈ 0.8000	m
Payload	Total amount of water carried by a drone (3 drops of 40 kg)	120	kg
Autonomy	Maximum distance a drone can travel between fuel refills	1000	m
ω_1	Inertia weight for first half of swarm	0.7298	
ω_2	Inertia weight for second half of swarm	0.9000	
v_{max}	Maximum drone velocity permitted	10	$\text{m}\cdot\text{s}^{-1}$

The model is implemented using Finite Differences with Dirichlet boundary conditions in space, and the fourth order Runge-Kutta method for the integration in time. Implicit methods are computationally too intensive for our needs. Fig. 2 shows an example of the model used for the prediction of the temperature profile and fuel energy density five minutes after four sparks have occurred. While very efficient, the simulation of this model using Matlab and a standard PC still runs about five times slower than real-time (without visualisation).

Once the model of the spread of wildfires runs realistically and reasonably fast, the model of the (collision-free) self-organising drones is coupled. In the first instance, we used a swarm of 50 drones which are launched from the bottom-left corner 20s after the sparks have occurred. The animations are very realistic, showing the drones moving between hotspots, with seemingly controlled ones coming back to life and going out of control while unattended. Snapshots of the temperature profile and of the fuel energy field at 100s and at 300s are shown in Fig. 3. As can be observed, the swarm of self-organising drones loses this fight.

Since this fire-fighting system is scalable, the fire can always be extinguished given sufficient time and a sufficient number of drones (at least for volume-less drones) –even if this number is unrealistically high. Hence

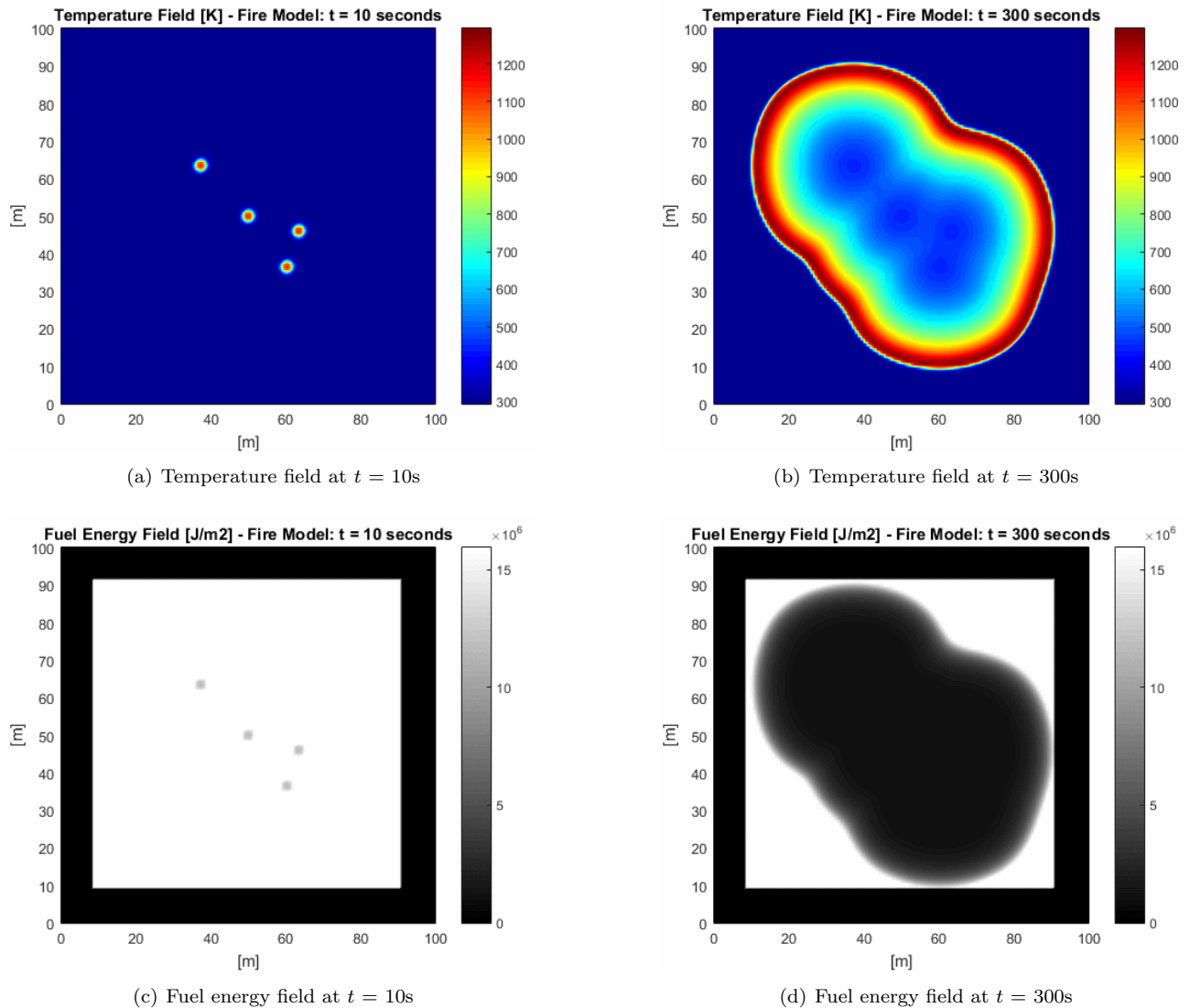


Figure 2: Fire-spread simulation: (a) Temperature field 10s after four sparks have occurred, (b) Temperature field after 300s have elapsed, (c) Fuel energy field 10s after four sparks have occurred, and (d) Fuel energy field after 300s have elapsed.

we simply doubled the size of the swarm and tested the simulation for 100 drones. Snapshots of the temperature profile and of the fuel energy field at 100s and at 300s are shown in Fig. 4. As can be observed, the swarm of self-organising drones wins this fight.

6 Conclusions and Future Work

This work comprises a proof of concept to investigate the feasibility and potential of using swarms of self-organising drones not only to assist human fire-fighters and to support decision-makers but also to autonomously and cooperatively strive to extinguish wildfires without human intervention. The numerical experiments are not aimed at reaching a conclusion as to whether they are able to put out real fires, since the models are still too simple and the settings rather arbitrary. Nonetheless, the experiments show that the self-organising decentralised cooperative behaviour is very powerful and a promising line of research to deal with such complex dynamic problem. The next step in our research is to incorporate collision-avoidance algorithms, to design and test other neighbourhood structures, and to engage more intensively in the development of self-organisation algorithms to achieve more sophisticated emergent fire-fighting behaviour than the one that results from simply searching for hotspots. For these studies, the fire-spread model needs to remain efficient. More advanced models

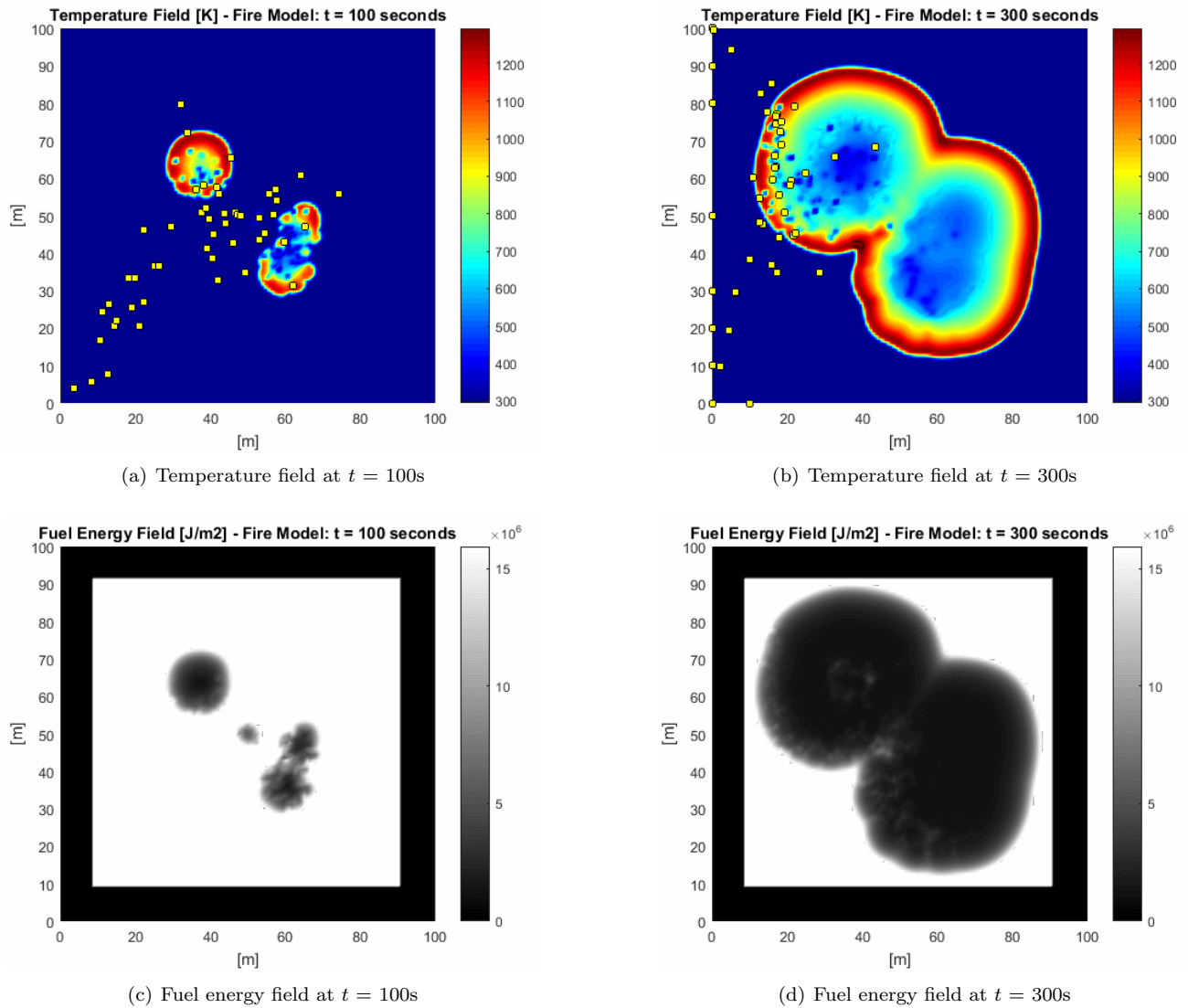


Figure 3: 50-drone self-organising swarm fighting the spread of a wildfire: (a) Temperature field 100s after four sparks have occurred, (b) Temperature field after 300s have elapsed, (c) Fuel energy field 100s after four sparks have occurred, and (d) Fuel energy field after 300s have elapsed.

of the spread of wildfires will be required for later testing of the most promising algorithms.

Acknowledgements

The authors would like to acknowledge the financial support of the Institute for Future Transport and Cities.

References

- [BB02] Tim M Blackwell and Peter J Bentley. Dynamic search with charged swarms. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pages 19–26. Morgan Kaufmann Publishers Inc., 2002.
- [BB06] T. Blackwell and J. Branke. Multiswarms, exclusion, and anti-convergence in dynamic environments. *IEEE Transactions on Evolutionary Computation*, 10(4):459–472, aug 2006.
- [CBM⁺05] David W Casbeer, Randal W Beard, Timothy W McLain, Sai-Ming Li, and Raman K Mehra. Forest fire monitoring with multiple small uavs. In *American Control Conference, 2005. Proceedings of the 2005*, pages 3530–3535. IEEE, 2005.

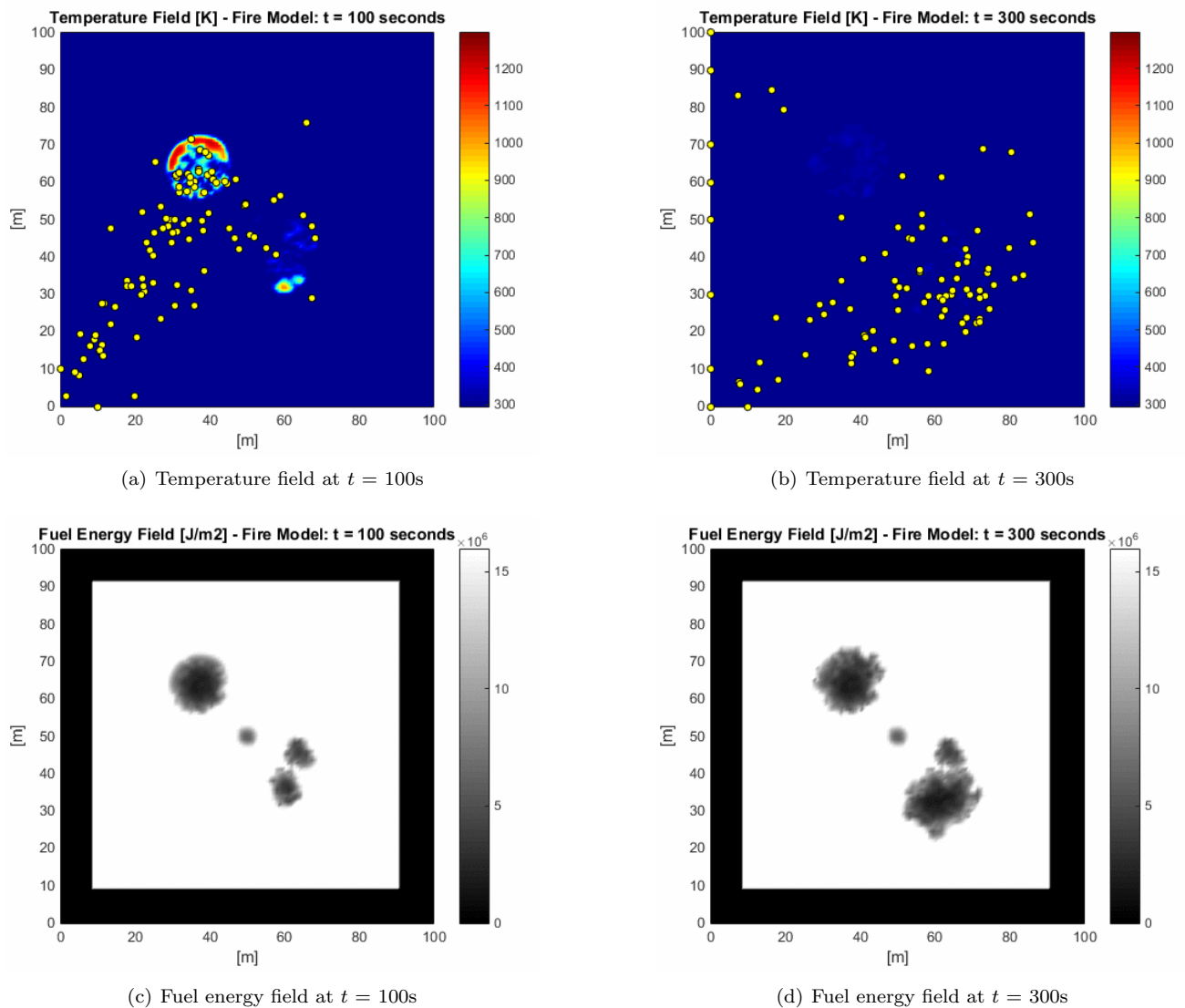


Figure 4: 100-drone self-organising swarm fighting the spread of a wildfire: (a) Temperature field 100s after four sparks have occurred, (b) Temperature field after 300s have elapsed, (c) Fuel energy field 100s after four sparks have occurred, and (d) Fuel energy field after 300s have elapsed.

- [CD00] Anthony Carlisle and Gerry Dozier. Adapting particle swarm optimization to dynamic environments. In *International conference on artificial intelligence*, volume 1, pages 429–434, 2000.
- [CJ10] Ying-ping Chen and Pei Jiang. Analysis of particle interaction in particle swarm optimization. *Theoretical Computer Science*, 411(21):2101–2115, 2010. *Swarm Intelligence Theory: A Snapshot of the State of the Art*.
- [CK02] Maurice Clerc and James Kennedy. The particle swarm – explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, feb 2002.
- [FAM07] L. Ferragut, M.I. Asensio, and S. Monedero. A numerical method for solving convection–reaction–diffusion multivalued equations in fire spread modelling. *Advances in Engineering Software*, 38(6):366–371, jun 2007.
- [Fin04] Mark A. Finney. FARSITE: Fire area simulator - Model development and evaluation. Technical report, U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2004.

- [GDK⁺12] Sayan Ghosh, Swagatam Das, Debarati Kundu, Kaushik Suresh, and Ajith Abraham. Inter-particle communication and search-dynamics of lbest particle swarm optimizers: An analysis. *Information Sciences*, 182(1):156–168, 2012. Nature-Inspired Collective Intelligence in Theory and Practice.
- [GW08] Ronald Graml and Grant Wigley. Bushfire hotspot detection through uninhabited aerial vehicles and reconfigurable computing. In *2008 IEEE Aerospace Conference*. IEEE, mar 2008.
- [IASD15] Mauro Sebastián Innocente, Silvana Maria Bastos Afonso, Johann Sienz, and Helen Margaret Davies. Particle swarm algorithm with adaptive constraint handling and integrated surrogate model for the management of petroleum fields. *Applied Soft Computing*, 34:463–484, sep 2015.
- [IS10] Mauro Sebastián Innocente and Johann Sienz. Coefficients’ settings in particle swarm optimization: Insight and guidelines. In Eduardo Dvorkin, Marcela Goldschmit, and Mario Storti, editors, *Mecom-Cilamce 2010*, volume XXIX, pages 9253–9269. Asociación Argentina de Mecánica Computacional (<http://www.amcaonline.org.ar>), November 2010.
- [IS11] Mauro Sebastián Innocente and Johann Sienz. Particle swarm optimization with inertia weight and constriction factor. In *Proceedings of the 2011 International conference on swarm intelligence (ICSI 2011)*, pages id-1–id-11, June 2011.
- [KE95] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, Nov/Dec 1995.
- [KES01] James Kennedy, Russell C. Eberhart, and Yuhui Shi. *Swarm intelligence*. The Morgan Kaufmann series in evolutionary computation. Morgan Kaufmann Publishers, 2001.
- [LAS18] Nandar Lynn, Mostafa Z. Ali, and Ponnuthurai Nagaratnam Suganthan. Population topologies for particle swarm optimization and differential evolution. *Swarm and Evolutionary Computation*, 39:24 – 35, 2018.
- [LBB06] Xiaodong Li, Jürgen Branke, and Tim Blackwell. Particle swarm with speciation and adaptation in a dynamic environment. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 51–58. ACM, 2006.
- [LE12] Barend J. Leonard and Andries P. Engelbrecht. Scalability study of particle swarm optimizers in dynamic environments. In *Lecture Notes in Computer Science*, pages 121–132. Springer Berlin Heidelberg, 2012.
- [Li04] Xiaodong Li. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In *Genetic and Evolutionary Computation Conference*, pages 105–116. Springer, 2004.
- [LY13] Changhe Li and Shengxiang Yang. A comparative study on particle swarm optimization in dynamic environments. In *Studies in Computational Intelligence*, pages 109–136. Springer Berlin Heidelberg, 2013.
- [MHM⁺13] Kevin McGrattan, Simo Hostikka, Randall McDermott, Jason Floyd, Craig Weinschenk, and Kristopher Overholt. Fire Dynamics Simulator technical reference guide volume 1: Mathematical model [6th Ed.]. Technical report, National Institute of Standards and Technology, 2013.
- [MSG02] J Margerit and O Séro-Guillaume. Modelling forest fires. part II: reduction to two-dimensional models and simulation of propagation. *International Journal of Heat and Mass Transfer*, 45(8):1723–1737, apr 2002.
- [PL06] D. Parrott and Xiaodong Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation*, 10(4):440–458, aug 2006.
- [PMG11] Hamid Parvin, Behrouz Minaei, and Sajjad Ghatei. A new particle swarm optimization for dynamic environments. In *Computational Intelligence in Security for Information Systems*, pages 293–300. Springer Berlin Heidelberg, 2011.

- [R⁺72] Richard C Rothermel et al. A mathematical model for predicting fire spread in wildland fuels. 1972.
- [Ric90] Gwynfor D Richards. An elliptical growth model of forest fire fronts and its numerical solution. *International Journal for Numerical Methods in Engineering*, 30(6):1163–1179, 1990.
- [RJR14] O. Rios, W. Jahn, and G. Rein. Forecasting wind-driven wildfires using an inverse modelling approach. *Natural Hazards and Earth System Science*, 14(6):1491–1503, jun 2014.
- [RMN11] Iman Rezazadeh, Mohammad Reza Meybodi, and Ahmad Naebi. Adaptive particle swarm optimization algorithm in dynamic environments. In *2011 Third International Conference on Computational Intelligence, Modelling & Simulation*. IEEE, sep 2011.
- [Şah05] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *Swarm Robotics*, pages 10–20. Springer Berlin Heidelberg, 2005.
- [SE98] Yuhui Shi and Russell C. Eberhart. A modified particle swarm optimizer. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (CEC 1998)*, pages 69–73, May 1998.
- [SGM02] O. Séro-Guillaume and J. Margerit. Modelling forest fires. part i: a complete set of equations derived by extended irreversible thermodynamics. *International Journal of Heat and Mass Transfer*, 45(8):1705–1722, apr 2002.
- [SPR15] Sadrollah Sadeghi, Hamid Parvin, and Farhad Rad. Particle swarm optimization algorithm for dynamic environments. In *Lecture Notes in Computer Science*, pages 260–269. Springer International Publishing, 2015.
- [TBC⁺17] Chiara Torresan, Andrea Berton, Federico Carotenuto, Salvatore Filippo Di Gennaro, Beniamino Gioli, Alessandro Matese, Franco Miglietta, Carolina Vagnoli, Alessandro Zaldei, and Luke Wallace. Forestry applications of uavs in europe: A review. *International Journal of Remote Sensing*, 38(8-10):2427–2447, 2017.
- [TBW⁺10] C Tymstra, RW Bryce, BM Wotton, SW Taylor, OB Armitage, et al. 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), 2010.
- [TSC⁺08] Kevin Tolhurst, Brett Shields, Derek Chong, et al. Phoenix: development and application of a bushfire risk management tool. *Australian Journal of Emergency Management, The*, 23(4):47, 2008.
- [UK-17] UK-RAS White Papers. Extreme environments robotics. Robotics for emergency response, disaster relief and resilience. Technical report, 2017.
- [XGZ18] Xuewen Xia, Ling Gui, and Zhi-Hui Zhan. A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting. *Applied Soft Computing*, 67:126 – 140, 2018.
- [YNSMM13] Danial Yazdani, Babak Nasiri, Alireza Sepas-Moghaddam, and Mohammad Reza Meybodi. A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization. *Applied Soft Computing*, 13(4):2144–2158, apr 2013.
- [YZL15] Chi Yuan, Youmin Zhang, and Zhixiang Liu. 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(7):783–792, 2015.