Stigmergy-based collision-avoidance algorithm for self-organising swarms

Paolo Grasso[1,2][0000−0003−1195−1859] and
Mauro Sebastián Innocente[1,3][0000−0001−8836−2839]

[1] Autonomous Vehicles & Artificial Intelligence Laboratory (AVAILab),
Centre for Future Transport and Cities, Coventry University, Coventry, UK
https://www.availab.org/
[2] grassop@coventry.ac.uk
[3] Mauro.S.Innocente@coventry.ac.uk

5th Int. Conference on Computational Vision and Bio-Inspired Computing

Abstract. Real-time multi-agent collision-avoidance algorithms comprise a key enabling technology for the practical use of self-organising swarms of drones. This paper proposes a decentralised reciprocal collision-avoidance algorithm, which is based on stigmergy and scalable. The algorithm is computationally inexpensive, based on the gradient of the locally measured dynamic cumulative signal strength field which results from the signals emitted by the swarm. The signal strength acts as a repulsor on each drone, which then tends to steer away from the *noisiest* regions (cluttered environment), thus avoiding collisions. The magnitudes of these repulsive forces can be tuned to control the relative importance assigned to collision avoidance with respect to the other phenomena affecting the agent's dynamics. We carried out numerical experiments on a self-organising swarm of drones aimed at fighting wildfires autonomously. As expected, it has been found that the collision rate can be reduced either by decreasing the cruise speed of the agents and/or by increasing the sampling frequency of the global signal strength field. A convenient by-product of the proposed collision-avoidance algorithm is that it helps maintain diversity in the swarm, thus enhancing exploration.

Keywords: Decentralised · Multi-agent · Autonomous · Wildfires.

1 Introduction

Multi-agent systems such as drone swarms consist of a multitude of decision-making individuals that interact directly and indirectly within the environment in order to achieve one or more predefined goals. In such systems, one of the most critical issues to be dealt with is collision avoidance. In this paper, we adhere to the classification of Collision-Avoidance Systems (CASs) suggested in [1] and [19], which identifies the following five classes:

1. Predefined collision avoidance.
2. Protocol-based collision avoidance.

3. Optimised escape trajectory approaches.
4. E-field / potential field methods.
5. Miscellaneous.

The first two categories encompass the original and most trivial approaches. Predefined collision avoidance follows a fixed set of unchangeable rules requiring no computation, whilst protocol-based collision avoidance relies on both a set of rules and continuous exchange of information among agents with regards to their states. The former results in a faster reaction time whereas the latter is safer. The most advanced CASs fall within the remaining three categories.

The optimised escape trajectory approach consists of solving an optimisation problem that combines the drone kinematic model with a set of physical and behavioural constraints. The following decentralised multi-agent algorithms fall within this group: 1) the Reciprocal Velocity Obstacles (RVO) and the derived Optimal Reciprocal Collision-Avoidance (ORCA) algorithms [2]; 2) the Cooperative Dynamic (CoDy) algorithm [15], which is able to solve dead-lock situations; 3) the Context-Aware Route Planning (CARP) algorithm [4, 13], which is a graph routing algorithm aimed at finding the shortest trajectory and avoiding collisions; and 4) other optimisation-based algorithms such as [14] and [20].

Among the E-field / potential field methods, the original ones tend to get drones randomly trapped in local minima (e.g. [11]), though this issue has now been dealt with (e.g. [6, 12]). The advantages are their low computational cost and short reaction time, whilst two major disadvantages still persist: 1) the requirement of high-level flight guidance, and 2) the potential occurence of hard-to-perform sharp discontinuities in the commanded manoeuvre.

The miscellaneous category includes approaches such as a *sense-and-avoid* algorithm for structured multi-agent systems, in which a leader is followed blindly by other agents [17, 19]. Other researchers explore more exotic solutions such as a two-stage reinforcement learning approach for multi-drone systems under imperfect sensing [18]. Their aim is to train a policy to plan a collision-free trajectory by leveraging local noisy observations. Others try to infer the state of the overall swarm through stigmergy, assuming this would be useful for collision avoidance purposes. For example, an approach based on anticipatory stigmergic collision avoidance (ASCA) under noise is proposed in [14], which consists of using pheromone information in a rather unusual fashion: instead of leaving a trail of pheromones over past positions, all drones in the swarm share information in the same indirect manner but about future intended positions instead. The drones will then optimise their trajectories in order to avoid locations with high concentrations of pheromone.

Numerous CASs have been proposed in the literature, the majority of which are suitable for Autonomous Ground Vehicles (AGVs) only. This is because CASs typically rely on two main assumptions [14]: 1) *stationary vehicles* (i.e. vehicles can stop suddenly and remain still indefinitely) and 2) *perfect information* (accurate noiseless perception). This is certainly not the case for Unmanned Aerial Vehicles (UAVs) or Unmanned Underwater Vehicles (UUVs), which might need to move with limited awareness of other vehicles' locations due to vision obstructions (e.g. smoke, cluttered environments, cloudy water) as well as high variance and bias of the positioning system being used.

In this paper, we propose a reactive decentralised stigmergy-based reciprocal collision-avoidance algorithm to be implemented on a self-organising swarm of drones. The algorithm is tested on a two-dimensional autonomous firefighting system based on [8, 10] aimed at suppressing a simulated wildfire [7]. Formulation allows for a relatively seamless generalisation to three dimensions, which is beyond the scope of this paper. The proposed multi-agent collision-avoidance algorithm was inspired by the E-field / potential field methods [1], the charged particles method [3], and the classical Reynold's rules [5, 16]. However, these methods are based on direct communication and inter-agent distance computations. Conversely, the proposed algorithm is indirect via stigmergy using information that is continuously available and dynamically changing in the environment: the cumulative strength of a continuously broadcasted signal emitted by every drone. Thus, the drone behavioural rules are modified by adding a repulsor that pushes the trajectory in the direction opposite to the signal gradient. Numerical experiments are performed in order to investigate the effects of *cruise speed* and *sampling frequency* on the performance of the algorithm. Finally, future work towards generalising the algorithm is suggested.

2 Multi-Agent Collision Avoidance Based on Stigmergy

A stigmergy-based multi-agent collision-avoidance algorithm is developed as an efficient and reliable alternative to multi-agent algorithms in the literature which require inter-agent direct communications. This approach is expected to be more reliable in real-world applications such as firefighting, since the required information is measured from –and in turn modifies– the environment therefore avoiding the loss of direct agent-to-agent communications. Given the early stages of this research, comparisons against other methods are not carried out in this paper. Instead, the aim is to demonstrate that the proposed method works and to investigate how *cruise speed* and *signal sampling time* may affect its performance.

Considering that the swarm-dynamics model used [8] is based on the Particle Swarm Optimisation method, the proposed collision-avoidance algorithm is embedded within the particle (i.e. agent or drone) trajectory difference equation in the form of a repulsor. Since each agent has no information regarding the location of the others, indirect communication (i.e. stigmergy) comprises a highly desirable feature: communication among agents is carried out by measuring and modifying the surrounding signal strength field in the environment. Thus, the process consists of two main stages: 1) sufficiently high frequency simultaneous sampling of the signal intensity in a few points surrounding the agent to calculate the gradient, and 2) correction of the attractor by a convex combination of the collision-avoidance repulsor and the original attractor.

The original formulation of the target position of every drone is regularly estimated at a predefined frequency as follows [8]:

$$\begin{cases} \mathbf{xt}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \omega \cdot \left(\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t-1)}\right) + \boldsymbol{\phi}_i^{(t)} \odot \left(\mathbf{p}_i^{(t)} - \mathbf{x}_i^{(t)}\right) \\ \boldsymbol{\phi}_i^{(t)} = \phi_{\min} + \mathbf{U}_{(\mathbf{0},\mathbf{1})} \cdot (\phi_{\max} - \phi_{\min}) \\ \phi_{\max}, \phi_{\min} = f(\omega) \end{cases} \tag{1}$$

where $\mathbf{xt}_i^{(t)}$ and $\mathbf{x}_i^{(t)}$ are the target and actual positions of the i^{th} drone, respectively, at time-step t; $\boldsymbol{\phi}_i^{(t)}$ is the acceleration coefficient; $\mathbf{p}_i^{(t)}$ is the attractor; ω is the inertia weight; $\mathbf{U}_{(\mathbf{0},\mathbf{1})} = \left[U_{(0,1)}, U_{(0,1)}\right]^{\mathrm{T}}$ with $U_{(0,1)}$ being a random variable realised from a uniform distribution within (0,1); and $\odot$ represents the component-wise product. The bounds $\phi_{\min} = (\omega + 1)$ and $\phi_{\max} = (\sqrt{\omega} - 1)^2$ are defined in [9] and [8] to ensure oscillatory behaviour of the drones.

Note that $\mathbf{x}_i^{(t+1)}$ is generally not equal to $\mathbf{xt}_i^{(t+1)}$ due to constraints such as the maximum cruise speed. It is also important to note that $\boldsymbol{\phi}_i^{(t)}$ takes different values for different drones, for different components, and for different time-steps because $U_{(0,1)}$ is resampled anew every time it is referenced. However, $U_{(0,1)}$ is replaced by a constant of 0.5 in this paper to obtain deterministic behaviour thus facilitating the study of other relevant parameters. If in recharging or water replenishment modes, the attractor is given by the location of the respective sources. In this paper, the attractor is modified by adding a repulsor associated to the signal strength field for collision avoidance purposes ($\mathbf{q}_i^{(t)}$ in (2)).

$$\begin{cases} \mathbf{p}_i^{(t)\star} = (1 - k_{\text{ca}}) \cdot \mathbf{p}_i^{(t)} + k_{\text{ca}} \cdot \mathbf{q}_i^{(t)} \\ 0 \leq k_{\text{ca}} < 1 \end{cases} \tag{2}$$

The repulsor $\mathbf{q}_i^{(t)}$ is function of the gradient of the signal strength field (σ):

$$\begin{cases} q_{ij}^{(t)} = x_{ij}^{(t)} - k_{\sigma} \cdot \frac{\partial \sigma(\mathbf{x}_i^{(t)})}{\partial x_j} \quad j = 1, 2 \\ \sigma(\mathbf{x}_i^{(t)}) = \mathit{\Sigma}_{m=1}^{N} \sigma_m(\mathbf{x}_i^{(t)}) \end{cases} \tag{3}$$

The individual signal intensity fields (σ_m) are normalised and defined as a power law based on the distance from the m^{th} source as in (4).

$$\begin{gathered} \sigma_m(\mathbf{x}_i^{(t)}) = \begin{cases} r_{\text{ref}}^2 \cdot r_{im}^{-2} & r_{im} > r_{\text{ref}} \\ 1 & 0 \leq r_{im} \leq r_{\text{ref}} \end{cases} \\ r_{im} = \left\| \mathbf{x}_i^{(t)} - \mathbf{x}_m^{(t)} \right\| \end{gathered} \tag{4}$$

where r_{ref} is a reference radius of a circle delimiting the area of generation of the signal within each drone. Setting r_{ref} approximately equal to the size of the drone or smaller ensures that the second condition in (4) never occurs. The proportionality constants in (2) and (3) depend on the magnitude of the other attractors in (1), and therefore need to be tuned for the problem in hand. In this paper, $k_{\text{ca}} = 0.7$ and $k_{\sigma} = 1000$. The gradient was calculated on a stencil of eight points surrounding the position of the drone on a structured grid with the same increment in both directions ($\Delta x = \Delta y$) as shown in (5).

$$\begin{cases} \dfrac{\partial \sigma}{\partial x_j} \simeq \dfrac{d\sigma}{dx_i} \\ \\ \dfrac{d\sigma}{dx_1} = \dfrac{1}{6\Delta x} \cdot (\sigma_{\mathrm{E}} - \sigma_{\mathrm{W}} + 0.5 \cdot (\sigma_{\mathrm{NE}} + \sigma_{\mathrm{SE}} - \sigma_{\mathrm{SW}} - \sigma_{\mathrm{NW}})) \\ \\ \dfrac{d\sigma}{dx_2} = \dfrac{1}{6\Delta y} \cdot (\sigma_{\mathrm{N}} - \sigma_{\mathrm{S}} + 0.5 \cdot (\sigma_{\mathrm{NE}} + \sigma_{\mathrm{NW}} - \sigma_{\mathrm{SW}} - \sigma_{\mathrm{SE}})) \end{cases} \tag{5}$$

3 Numerical Experiments

The stigmergy-based multi-agent collision-avoidance algorithm developed in this paper is tested on a swarm of deterministically self-organising drones [8] fighting a simulated wildfire [7]. The collision-avoidance algorithm is always active while drones are airborne during any of the following mission phases for the simulated firefighting operations [8] in Figure 1:

- *Initialisation* Drones are positioned in their docking station with no water in their hull, the combustible vegetation fuel is distributed in the domain, and the wildfire is started via multiple ignition points (e.g. three in Fig. 1).
- *Firefighting* Drones search for fires by recording their best experience and sharing it with the entire swarm. They will pour a predefined amount of water when flying over their target position.
- *Water collection* When the water payload is less than 30% of full capacity, the drone's target is changed to the water reservoir location (top left corner).
- *Recharging* When the battery charge is low, the drone's target is changed to the location of the docking/charging station (bottom left corner).
- *Check* After all fires are believed to be suppressed, the swarm will explore the domain searching for new hot spots.
- *Landing* The nearest (six) drones have priority to fly towards their original docking stations and land. When landed, the collision-avoidance algorithm is deactivated and their positions will not be considered any more for the collision count.
- *Hovering* The farthest drones will fly towards a waiting area, hovering until there is no more drones with higher priority for landing.
- *Landed* After every drone has landed, the simulation will continue for another minute to check that all fires are properly extinguished and that no re-ignition occurs.

While the first and the last phases are fixed, the others are change dynamically and can coexist. That is to say, some drones may be suppressing fires while others are flying to replenish water or recharge batteries.

Various numerical experiments have been performed with different combinations of maximum cruise speed (v) and sampling frequency of the signal field (f). For each experiment, the total number of collisions (C) and the duration of the mission (T) were recorded. A few representative cases are provided in Table 1. It is important to note that every drone survives any collision in these numerical experiments. Although this may sound unrealistic, it is done puposely because it is useful for statistical purposes: the rate of collisions would decrease due to a shrinking swarm rather than due to a successful collision-avoidance algorithm.

Table 1: Experimental results for various combinations of cruise speed (v) and sampling frequency (f), where C is the total collision count and T is the duration of the simulation.

ID	v [m·s^{-1}]	f [Hz]	C [-]	T [s]	$f \cdot v^{-1}$ [m^{-1}]	$C \cdot T^{-1}$ [s^{-1}]
1	5	30	0	551	6.0	0.00
2	10	30	7	264	3.0	0.03
3	15	30	244	182	2.0	1.34
4	20	30	1019	169	1.5	6.03
5	20	40	243	167	2.0	1.46
6	20	50	36	159	2.5	0.12
7	30	80	1	127	2.7	0.01

Experiments 1 to 4 were carried out with the same sampling frequency (30 Hz), showing that increasing cruise speed (5, 10 and 15 m/s) exponentially deteriorates the performance of the proposed collision-avoidance algorithm. In turn, experiments 4 to 6 were carried out with the same cruise speed (20 m/s), showing that increasing sampling frequency (30, 40 and 50 Hz) dramatically decreases the collision count. Whilst the available computational resources allowed us to carry out experiments up to $f = 80$ Hz, it is reasonable to expect that $f = 100$ Hz would bring the collision rate to zero in this particular case. Further experiments will be performed in the near future to support this claim.

By studying the values of the frequency-to-velocity and collisions-to-time ratios (last two columns in Table 1), we identify a hyperbolic trend: the collisions-to-time ratio tends to infinity for decreasing frequency-to-velocity ratios, whilst the collisions-to-time ratio tends to zero for increasing frequency-to-velocity ratios. It is hypothesised that there is a function that describes the relation between these two coefficients. This is in line with the results from experiments 3 and 5 carried out with the same frequency-to-velocity ratio ($f \cdot v^{-1}$) and returning similar collisions-to-time ratios ($C \cdot T^{-1}$). Further experiments will be carried out in order to better investigate this hypothesis.

4 Conclusions

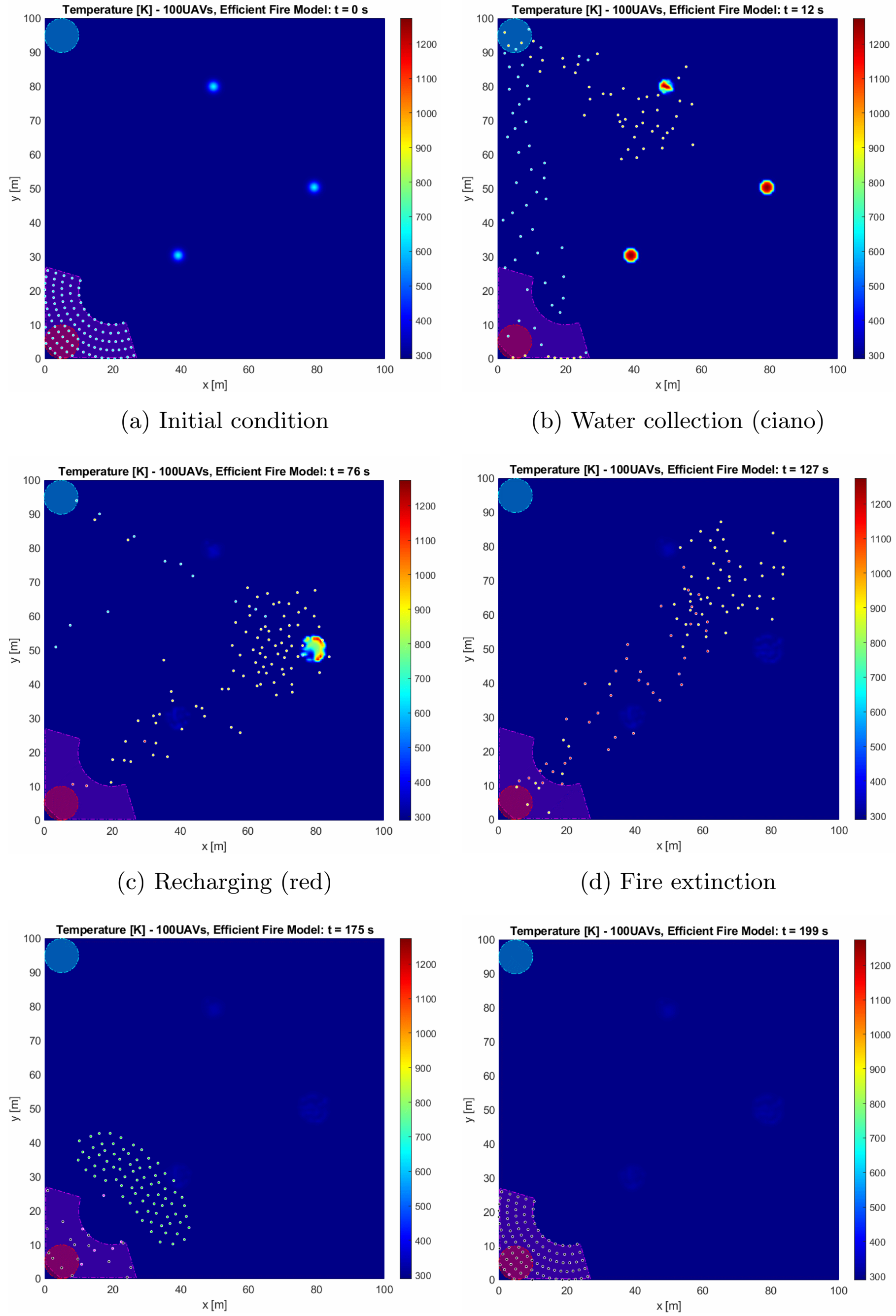
In this paper, we proposed and developed a reactive multi-agent decentralised stigmergy-based algorithm for reciprocal collision-avoidance. Its potential was demonstrated by implementing it on a swarm of drones self-organised to fight the propagation of simulated wildfires. Arguably, the algorithm could also be used for blind flight –i.e. when the drones have no means of knowing the positions of others. The effects of two main parameters were investigated, namely the cruise speed and the sampling frequency, reaching the expected conclusion that decreasing the former and/or increasing the latter results in the reduction of the collision rate. Furthermore, by analysing the frequency-to-speed and collisions-to-time ratios, we hypothesised that there might be a trend that should be further investigated. Furthermore, a convenient by-product of the proposed collision-avoidance algorithm has been observed: it helps maintain diversity in the swarm, thus enhancing exploration.

Purposely, no uncertainty in the environment, no stochasticity in the algorithms, and no flight dynamics were considered in order to focus on the impact of the stigmergy-based collision-avoidance algorithm on the collective behaviour and performance of the swarm. Future work will introduce these aspects as well as extend the developments to three-dimensional space.

References

1. Albaker, B.M., Rahim, N.A.: Unmanned aircraft collision detection and resolution: Concept and survey. In: 2010 5th IEEE Conference on Industrial Electronics and Applications. IEEE (jun 2010). https://doi.org/10.1109/iciea.2010.5516808
2. van den Berg, J., Lin, M., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: 2008 IEEE International Conference on Robotics and Automation. IEEE (may 2008). https://doi.org/10.1109/robot.2008.4543489
3. Blackwell, T.M., Bentley, P.J.: Dynamic search with charged swarms. In: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation. p. 19–26. GECCO'02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2002)
4. Blesing, C., Luensch, D., Stenzel, J., Korth, B.: Concept of a multi-agent based decentralized production system for the automotive industry. In: Lecture Notes in Computer Science, pp. 19–30. Springer International Publishing (2017), 10.1007/978-3-319-59930-4_2
5. Braga, R.G., da Silva, R.C., Ramos, A.C.B., Mora-Camino, F.: Collision avoidance based on reynolds rules: A case study using quadrotors. In: Latifi, S. (ed.) Information Technology - New Generations. Advances in Intelligent Systems and Computing. vol. 558, pp. 773–780. Springer International Publishing, Cham (2018)
6. Chang, D.E., Shadden, S., Marsden, J., Olfati-Saber, R.: Collision avoidance for multiple agent systems. In: 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475). IEEE. https://doi.org/10.1109/cdc.2003.1272619
7. Grasso, P., Innocente, M.S.: Physics-based model of wildfire propagation towards faster-than-real-time simulations. Computers & Mathematics with Applications **80**(5), 790–808 (sep 2020). https://doi.org/10.1016/j.camwa.2020.05.009

8. Innocente, M.S., Grasso, P.: Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems. Journal of Computational Science **34**, 80–101 (may 2019). https://doi.org/10.1016/j.jocs.2019.04.009
9. Innocente, M.S.: Particle swarms reformulated towards a unified and flexible framework. In: Tan, Y., Shi, Y. (eds.) Advances in Swarm Intelligence. ICSI 2021. Lecture Notes in Computer Science. vol. 12689, pp. 275–286. Springer Nature (2021)
10. Innocente, M.S., Grasso, P.: Swarms of autonomous drones self-organised to fight the spread of wildfires. In: Proceedings of the GEOSAFE Workshop on Robust Solutions for Fire Fighting (2018)
11. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings. 1985 IEEE International Conference on Robotics and Automation. Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/robot.1985.1087247
12. Kim, J.O., Khosla, P.: Real-time obstacle avoidance using harmonic potential functions. IEEE Transactions on Robotics and Automation **8**(3), 338–349 (jun 1992). https://doi.org/10.1109/70.143352
13. ter Mors, A.W.: Conflict-free route planning in dynamic environments. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE (sep 2011). https://doi.org/10.1109/iros.2011.6094461
14. von der Osten, F.B., Kirley, M., Miller, T.: Anticipatory stigmergic collision avoidance under noise. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. ACM (jul 2014). https://doi.org/10.1145/2576768.2598389
15. Regele, R., Levi, P.: Cooperative multi-robot path planning by heuristic priority adjustment. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE (oct 2006). https://doi.org/10.1109/iros.2006.282480
16. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87. ACM Press (1987). https://doi.org/10.1145/37401.37406
17. Soriano, A., Bernabeu, E.J., Valera, A., Vallés, M.: Multi-agent systems platform for mobile robots collision avoidance. In: Advances on Practical Applications of Agents and Multi-Agent Systems, pp. 320–323. Springer Berlin Heidelberg (2013), `10.1007/978-3-642-38073-0_37`
18. Wang, D., Fan, T., Han, T., Pan, J.: A two-stage reinforcement learning approach for multi-UAV collision avoidance under imperfect sensing. IEEE Robotics and Automation Letters **5**(2), 3098–3105 (apr 2020). https://doi.org/10.1109/lra.2020.2974648
19. Yasin, J.N., Mohamed, S.A.S., Haghbayan, M.H., Heikkonen, J., Tenhunen, H., Plosila, J.: Navigation of autonomous swarm of drones using translational coordinates. In: Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection, pp. 353–362. Springer International Publishing (2020), `10.1007/978-3-030-49778-1_28`
20. Zhang, X., Liniger, A., Borrelli, F.: Optimization-based collision avoidance (Nov 2017)



(a) Initial condition

(b) Water collection (ciano)

(c) Recharging (red)

(d) Fire extinction

(e) Hovering (green) & landing (magenta)

(f) Landed (black) on docking station

Fig. 1: Top views of a 100m×100m homogeneous distribution of combustible vegetation ignited in three locations (Fig. a) and some representative instances for the main mission phases of a swarm of 100 firefighting drones. The colourmap represents the temperature field in Kelvin, ranging from ambient temperature (dark blue) to high flaming temperature (dark red). The red circle on the bottom-left of each figure shows the recharging area, the blue circle on the top-left corner shows the water source, whilst the *arena polygon* shows the landing area.