Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 373 (2021) 113446
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A digital twin framework for machine learning optimization of aerial fire fighting and pilot safety

T.I. Zohdi

Department of Mechanical Engineering 6195 Etcheverry Hall, University of California, Berkeley, CA, 94720-1740, USA

Received 26 July 2020; received in revised form 11 September 2020; accepted 13 September 2020
Available online 5 October 2020

Abstract

The objective of this work is to model and simulate aerial drops of fire retardants in dangerous fire environments. Specifically, the work develops a computational framework for a model problem combining:

1. A meshless discrete element component that tracks the trajectory of released airborne materials from a controlled aircraft, ranging from retardant powders to encapsulated packets, subjected to prevailing wind velocities and fire-driven updrafts.
2. A Machine Learning Algorithm (MLA) to rapidly ascertain the optimal aircraft (unmanned or manned) dynamics to maximize the fire-retardant release effectiveness (released material usage and target impact).

The framework is designed to enable Digital Twin type technologies, i.e. digital replicas that run in real time with the physical system. However, it is also designed to run at much faster rates, in order to enable MLA's to optimize the planning, by running quickly on laptops and mobile systems. The overall guiding motivation is to provide a useful tool to enable rapid flight-path planning for aerial first-responders *in real-time* and to train pilots. Numerical examples are provided to illustrate the process.
© 2020 Elsevier B.V. All rights reserved.

Keywords: Aerial fire-fighting; Fire retardants; Optimization; Machine-learning

1. Introduction

Aerial fire-fighting is a term used for the employment of aircraft (fixed wing, helicopters, etc.) in fighting fires to deliver water, fire-retardants, etc. There are a variety of chemicals in use (US Department of Agriculture [1,2]), such as water-based mixtures, foams, gels and commercial products like Phos-Chek, which are manufactured as dry powders or as concentrated liquids and diluted with water before deployment, consisting of ammonium polyphosphate, diammonium phosphate, diammonium sulfate, monoammonium phosphate, attapulgus clay, guar gum, etc. The idea of aerial fire-fighting started to be taken quite seriously in the 1930's when aircraft had sufficiently evolved. The most common use is that of a tanker or high-volume transport that can be quickly loaded with fire retardants. There are, of course, many variants for rapid loading, such as water scoopers, large helicopters, etc. The general term for such aircraft is "water-bomber" (Fig. 1). Various aircraft have been used over the years for firefighting [3–6]. For example:

E-mail address: zohdi@berkeley.edu.

https://doi.org/10.1016/j.cma.2020.113446
0045-7825/© 2020 Elsevier B.V. All rights reserved.

Fig. 1. An aerial water bomber. *Source:* Courtesy of the public domain website: https://pixabay.com.

- **Small-sized:** The smallest aircraft used are the Single Engine Air Tankers (SEATs), which are agricultural sprayers that drop approximately 3000 liters of retardant. These include the Air Tractor AT-802 and the Soviet Antonov An-2 biplane.
- **Medium-sized:** Medium-sized are often modified aircraft, such as the Grumman S-2 Tracker (retrofitted with turboprop engines as the S-2T) as used by the California Department of Forestry and Fire Protection, as well as the Conair Firecat version developed and used by Conair Group Inc. of Canada, and the converted Convair 580 and Fokker F27 Friendship turboprop airliners to air tankers.
- **Large-sized:** The largest aerial firefighter ever used is a modified Boeing 747 known as the Global Supertanker, which carries 74,200 liters fed by a pressurized drop system. It was used first in 2009 in Spain. Another large aircraft used is the Russian convertible-to-cargo Ilyushin Il-76 airtankers that can carry 41,600 liters.

As of 2020, there are literally hundreds of aircraft models in use. During operations, an Air Tactical Group Supervisor (ATGS) is typically in charge of managing an array of assets assigned to the fire and may be oftentimes charged with the first response or extended operations. The course of action is complex and requires low flyovers, rapid turns, etc., with huge updrafts and low visibility. Furthermore, depending on the size and type of fire, it may require coordination with ground crews and other aircraft. *These operations can be quite dangerous, with many pilots having lost their lives over the last 40 years.* We refer the reader to Christopher [7], Gabbert [8], Jendsch [9], NTSB [10,11] for more details. There are additional proposed ways of reducing risk. For example, because it is very difficult to safely precisely target fluidized retardants from the air, it has been proposed to employ encapsulated retardant "bomblets" or compressed powder packets. The targeting can be to establish a perimeter around the fire or to achieve a direct assault on the fire itself. There are a variety of encapsulated fire-retardant products available which activate/release depending on the ambient thermal environment (Sanford [12], Boeing [13,14], Limer [15] and Chang [16]). Variants have been proposed using bombing techniques used in warfare, in particular, ironically, incendiary warfare. It is no coincidence that the term "air attack" is used in industry to describe the tactics and maneuvers needed in aerial fire-fighting. Such maneuvers require extremely sophisticated piloting skills.

Generally, much of the corresponding fire-fighting research is geared towards improving emergency response times and practices. In particular, within the realm of simulation technologies, while large-scale computational simulations of fire-related events exist, they are oftentimes too slow to be useful for field-deployed, fast-paced, mobile computing platforms in harsh environments. Accordingly, the objective of this work is to develop rapidly computable digital-twins of aerial fire-fighting systems, which are digital replicas of physical systems that blend artificial intelligence, machine learning, and software analytics to create living computer models that change in *tandem* with their physical counterparts. Specifically, the goal is to develop a system capable of training pilots,

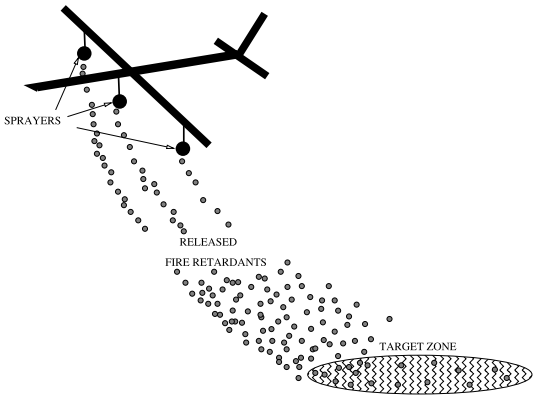

Fig. 2. A schematic of the model problem.

programming Unmanned Aerial Vehicles (UAVs) and path planning, in a virtual setting, in order to reduce risk and losses during aerial fire-fighting. The specific objective is to simulate and optimize aircraft dynamics for maximum effectiveness in delivering fire retardants. The work develops a computational framework for a model problem combining:

1. A meshless discrete element component that tracks the trajectory of released airborne materials from a controlled aircraft, ranging from retardant powders to encapsulated packets, subjected to prevailing wind velocities and fire-driven updrafts,
2. A Machine Learning Algorithm (MLA) to rapidly ascertain the optimal aircraft (unmanned or manned) dynamics to maximize the fire-retardant release effectiveness (released material usage and target impact).

The framework is designed to enable rapidly computable Digital Twin type technologies. However, it is also designed to run at much faster rates, in order to enable MLA's to optimize the planning, by running quickly on laptops and mobile systems. The overall guiding motivation is to provide a useful tool to enable rapid flight-path planning for aerial first-responders *in real-time* and to train pilots. Numerical examples are provided to illustrate the process. This research is a natural extension to recent work on fire propagation simulation found in Zohdi [17] and can be applied to a variety of new hybrid fire testing methods (Schulthess et al. [18]).

2. Model problem

We consider the release of a random distribution of packets into an ambient atmosphere (Fig. 2). We assume that the packets are small enough, relative to the scale of the overall problem, that they can be considered as particles.

Following formulations for physically similar problems associated with particulate dynamics from the fields of blasts, explosions and fire embers (Zohdi [19–21,17]), we make the following assumptions:

- We assume the same initial velocity magnitude for all particles under consideration, with a random distribution of outward directions away from the source of the retardant. This implies that a particle non-interaction approximation is appropriate. Thus, the inter-particle collisions are negligible. This has been repeatedly verified by "brute-force" collision calculations using formulations found in Zohdi [22–25].
- We assume that the particles are spherical with a random distribution of radii R_i, $i = 1, 2, 3 \ldots N = particles$. The masses are given by $m_i = \rho_i \frac{4}{3}\pi R_i^3$, where ρ_i is the density of the particles.
- We assume that the retardant particles are *quite small* and that the amount of rotation, if any, contributes negligibly to the overall trajectory of the particles. The equation of motion for the ith particle in the system is

$$m_i \dot{\boldsymbol{v}}_i = \boldsymbol{\Psi}_i^{grav} + \boldsymbol{\Psi}_i^{drag}, \tag{2.1}$$

with initial velocity $\boldsymbol{v}_i(0)$ and initial position $\boldsymbol{r}_i(0)$. The gravitational force is $\boldsymbol{\Psi}_i^{grav} = m_i \boldsymbol{g}$, where $\boldsymbol{g} = (g_x, g_y, g_z) = (0, -9.81, 0)\,\mathrm{m/s^2}$.

- For the drag, we will employ a general phenomenological model

$$\boldsymbol{\Psi}_i^{drag} = \frac{1}{2}\rho_a C_D \|\boldsymbol{v}^f - \boldsymbol{v}_i\|(\boldsymbol{v}^f - \boldsymbol{v}_i)A_i, \tag{2.2}$$

where C_D is the drag coefficient, A_i is the reference area, which for a sphere is $A_i = \pi R_i^2$, ρ_a is the density of the ambient fluid environment and $\boldsymbol{v}^f$ is the velocity of the surrounding medium which, in the case of interest, is air. We will assume that the velocity of the surrounding fluid medium ($\boldsymbol{v}^f$) is given, implicitly assuming that the dynamics of the surrounding medium are unaffected by the particles.[1]

In order to gain insight, initially, we will discuss the closely related, analytically tractable, Stokesian model, next.

Remark 1. As mentioned, there are a large number of physically similar phenomena to a retardant drop, such as the particulate dynamics associated with blasts, explosions and fire embers. We refer the interested reader to the wide array of literature on this topic; see Plimpton [26], Brock [27], Russell [28], Shimanzu [29], Werrett [30], Kazuma [31,32], Wingerden et al. [33] and Fernandez-Pello [34], Pleasance and Hart [35], Stokes [36] and Rowntree and Stokes [37], Hadden et al. [38], Urban et al. [39] and Zohdi [17] .

3. Analytical characterization: simplified Stokesian model

3.1. Analysis of particle velocities

For a (low Reynolds number) Stokesian model, the differential equation for each particle is (Fig. 2)

$$m_i \frac{d\boldsymbol{v}_i}{dt} = m_i \boldsymbol{g} + c_i(\boldsymbol{v}^f - \boldsymbol{v}_i) \tag{3.1}$$

where $c_i = \mu_f 6\pi R_i$, where μ_f is the viscosity of the surrounding fluid (air) and the local Reynolds number for a particle is $Re \stackrel{\text{def}}{=} \frac{2R_i\rho_a\|\boldsymbol{v}^f - \boldsymbol{v}_i\|}{\mu_f}$ and μ_f is the fluid viscosity. This can be written in normalized form as

$$\frac{d\boldsymbol{v}_i}{dt} + \underbrace{\frac{c_i}{m_i}}_{a_i}\boldsymbol{v}_i = \underbrace{\boldsymbol{g} + \frac{c_i}{m_i}\boldsymbol{v}^f}_{\boldsymbol{b}_i}. \tag{3.2}$$

This can be solved analytically to yield, for example in the y direction

$$v_{iy}(t) = \underbrace{(v_{iyo} - \frac{b_{iy}}{a_{iy}})}_{A_{iy}} e^{-\frac{c_i}{m_i}t} + \underbrace{\frac{b_{iy}}{a_{iy}}}_{B_{iy}}, \tag{3.3}$$

where

- $a_{iy} = \frac{c_i}{m_i} = \frac{9\mu_f}{2\rho_i R_i^2}$,
- $b_{iy} = g_y + \frac{c_i}{m_i} = g_y + \frac{9\mu_f}{2\rho_i R_i^2}$,
- $A_{iy} = v_{iyo} - (g_y \frac{2\rho_i R_i^2}{9\mu_f} + v_y^f)$,
- $B_{iy} = (g_y \frac{2\rho_i R_i^2}{9\mu_f} + v_y^f)$,

where the similar expressions hold for the x and z directions. The trends are

- As $t \to \infty$

$$v_{iy}(t = \infty) \to \frac{2g_y\rho_i R_i^2}{9\mu_f} + v_y^f, \tag{3.4}$$

[1] We will discuss these assumptions further, later in the work.

- As $R_i \to 0$

$$v_{iy}(t=\infty) \to v_y^f. \tag{3.5}$$

- The decay rate is controlled by $\frac{c_i}{m_i} = \frac{9\mu_f}{2\rho_i R_i^2}$, indicating that small particles attain ambient velocities extremely quickly.

Some special cases:

- With no gravity:

$$v_{iy}(t) = (v_{iyo} - v_y^f)e^{-\frac{c_i}{m_i}t} + v_y^f. \tag{3.6}$$

- With no damping:

$$\frac{dv_{iy}}{dt} = g_y \Rightarrow v_{iy}(t) = v_{iyo} + g_y t. \tag{3.7}$$

Again, we note that the equations are virtually the same for the x and z directions, with the direction of gravity and fluid flow being the main differentiators.

3.2. Analysis of particle positions

From the fundamental equation, relating the position $\boldsymbol{r}_i$ to the velocity

$$\frac{d\boldsymbol{r}_i}{dt} = \boldsymbol{v}_i, \tag{3.8}$$

we can write for the y direction

$$\frac{dr_{iy}}{dt} = v_{iy} = A_{iy}e^{-\frac{c_i}{m_i}t} + B_{iy} \tag{3.9}$$

with the similar relations x and z directions. Integrating and applying the initial conditions yields

$$r_{iy}(t) = r_{iyo} + \frac{m_i}{c_i}A_{iy}(1 - e^{-\frac{c_i}{m_i}t}) + B_{iy}t. \tag{3.10}$$

If $g_y = 0$ and $v_y^f = 0$, then

$$r_{iy}(t) = r_{iyo} + v_{iyo}2\rho_i\frac{R_i^2}{9\mu_f}(1 - e^{-\frac{9\mu_f}{2\rho_i R_i^2}t}). \tag{3.11}$$

As $t \to \infty$

$$r_{iy}(\infty) = r_{iyo} + v_{iyo}2\rho_i\frac{R_i^2}{9\mu_f}. \tag{3.12}$$

As $R_i \to 0$, the travel distance is dramatically shorter. The converse is true, larger particles travel farther.

3.3. Settling (airborne) time

The settling, steady-state velocity can be obtained directly from

$$\frac{d\boldsymbol{v}_i}{dt} + a_i\boldsymbol{v}_i = \boldsymbol{b}_i, \tag{3.13}$$

by setting $\frac{d\boldsymbol{v}_i}{dt} = \boldsymbol{0}$, one can immediately solve for the steady-state velocity

$$\boldsymbol{v}_i(\infty) = \frac{\boldsymbol{b}_i}{a_i} = \frac{2\rho_i R_i^2}{9\mu_f}\boldsymbol{g} + \boldsymbol{v}^f. \tag{3.14}$$

The trends are

- As $R_i \rightarrow 0$, then $\boldsymbol{v}_i(\infty) \rightarrow \boldsymbol{v}^f$,
- As $\boldsymbol{v}^f \rightarrow \boldsymbol{0}$, then $\boldsymbol{v}_i(\infty) \rightarrow \frac{2\rho_i R_i^2}{9\mu_f}\boldsymbol{g}$.

In summary

- Large particles travel far and settle quickly and
- Small particles do not travel far and settle slowly.

Remark 2. The ratio of the Stokesian drag force to gravity is

$$\frac{\| \boldsymbol{\Psi}^{drag,Stokesian} \|}{\| \boldsymbol{\Psi}^{grav} \|} = \frac{9\mu_f \| \boldsymbol{v}^f - \boldsymbol{v}_i \|}{2\rho_i R_i^2 g}, \tag{3.15}$$

which indicates that for very small particles, drag will dominate the settling process and for larger particles, gravity will dominate.

4. Computational approaches for more complex models

4.1. More detailed characterization of the drag

In order to more accurately model the effects of drag, one can take into account that the empirical drag coefficient varies with Reynolds number. For example, consider the following piecewise relation (Chow [40]):

- For $0 < Re \leq 1$, $C_D = \frac{24}{Re}$,
- For $1 < Re \leq 400$, $C_D = \frac{24}{Re^{0.646}}$,
- For $400 < Re \leq 3 \times 10^5$, $C_D = 0.5$,
- For $3 \times 10^5 < Re \leq 2 \times 10^6$, $C_D = 0.000366 Re^{0.4275}$ and
- For $2 \times 10^6 < Re < \infty$, $C_D = 0.18$,

where, as in the previous section, the local Reynolds number for a particle is $Re \stackrel{\text{def}}{=} \frac{2R_i \rho_a \| \boldsymbol{v}^f - \boldsymbol{v}_i \|}{\mu_f}$ and μ_f is the fluid viscosity.[2] *We note that in the zero Reynolds number limit, the drag is Stokesian.* In order to solve the governing equation,

$$m_i \dot{\boldsymbol{v}}_i = \boldsymbol{\Psi}_i^{grav} + \boldsymbol{\Psi}_i^{drag} = m_i \boldsymbol{g} + \frac{1}{2}\rho_a C_D \| \boldsymbol{v}^f - \boldsymbol{v}_i \| (\boldsymbol{v}^f - \boldsymbol{v}_i) A_i, \tag{4.1}$$

we integrate the velocity numerically

$$\boldsymbol{v}_i(t + \Delta t) = \boldsymbol{v}_i(t) + \frac{1}{m_i}\int_t^{t+\Delta t} (\boldsymbol{\Psi}_i^{grav} + \boldsymbol{\Psi}_i^{drag})\, dt \approx \boldsymbol{v}_i(t) + \frac{\Delta t}{m_i}\left(\boldsymbol{\Psi}_i^{grav}(t) + \boldsymbol{\Psi}_i^{drag}(t) \right). \tag{4.2}$$

The position is the obtained by integrating again:

$$\boldsymbol{r}_i(t + \Delta t) = \boldsymbol{r}_i(t) + \int_t^{t+\Delta t} \boldsymbol{v}_i(t)\, dt \approx \boldsymbol{r}_i(t) + \Delta t \boldsymbol{v}_i(t). \tag{4.3}$$

This approach has been used repeatedly for a variety of physically similar drift-type problems in Zohdi [19–21,17].

Remark 3. The piecewise drag law of Chow [40] is a mathematical description for the Reynolds number over a wide range and is a curve-fit of extensive data from Schlichting [41].

5. Modeling aircraft dynamics

5.1. Fuselage dynamics

We assume two control variables: (1) The angular velocity of the aircraft and (2) its speed. The relative velocity of any point p on the aircraft with respect to the center c is (Fig. 3)

$$\boldsymbol{v}_{c \rightarrow p} = \boldsymbol{\omega} \times \boldsymbol{r}_{c \rightarrow p}. \tag{5.1}$$

[2] The viscosity coefficient for air is $\mu_f = 0.000018$ Pa-s.

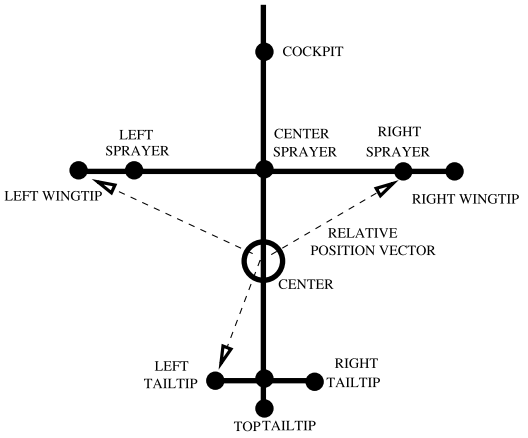

Fig. 3. The fuselage layout and position vectors to various locations. The entire body is governed by rigid body kinematics with two variables: the angular velocity of the body ($\boldsymbol{\omega}$) and the velocity of the center ($\boldsymbol{v}_c$). The velocities and positions of all other points can be determined by rigid body kinematics and integration.

The total velocity is

$$\boldsymbol{v}_p = \boldsymbol{v}_c + \boldsymbol{v}_{c\to p}. \tag{5.2}$$

Consequently, the new position is related to the velocity by

$$\frac{d\boldsymbol{r}_p}{dt} = \boldsymbol{v}_p \tag{5.3}$$

and thus

$$\boldsymbol{r}_p(t+\Delta t) = \boldsymbol{r}_p(t) + \int_t^{t+\Delta t} \boldsymbol{v}_p\, dt \approx \boldsymbol{r}_p(t) + \boldsymbol{v}_p(t)\Delta t \tag{5.4}$$

For this equation, we can ascertain any point's position relative to that of the center.

5.2. Flow rates for particle release

The total volume of particles dropped is given by

$$V_p = \zeta(T_F - T_I), \tag{5.5}$$

where ζ is the volume dropped per unit time, T_I is the time when the particle drops are initiated and T_F is the time when the particle drops stop. The radius of a particle is a (volume) control variable, R_i, with mass of the particle is given by $m_i = \rho_i \frac{4}{3}\pi R_i^3$. The number of particles is given by

$$N_p = \frac{\zeta(T_F - T_I)}{\frac{4}{3}\pi R_i^3}. \tag{5.6}$$

5.3. A numerical example

In order to illustrate the model, the following simulation parameters were chosen:

- Starting height of 200 m,
- Total simulation duration, 2.5 s,
- The time step size, $\Delta t = 5 \times 10^{-2}$ s,

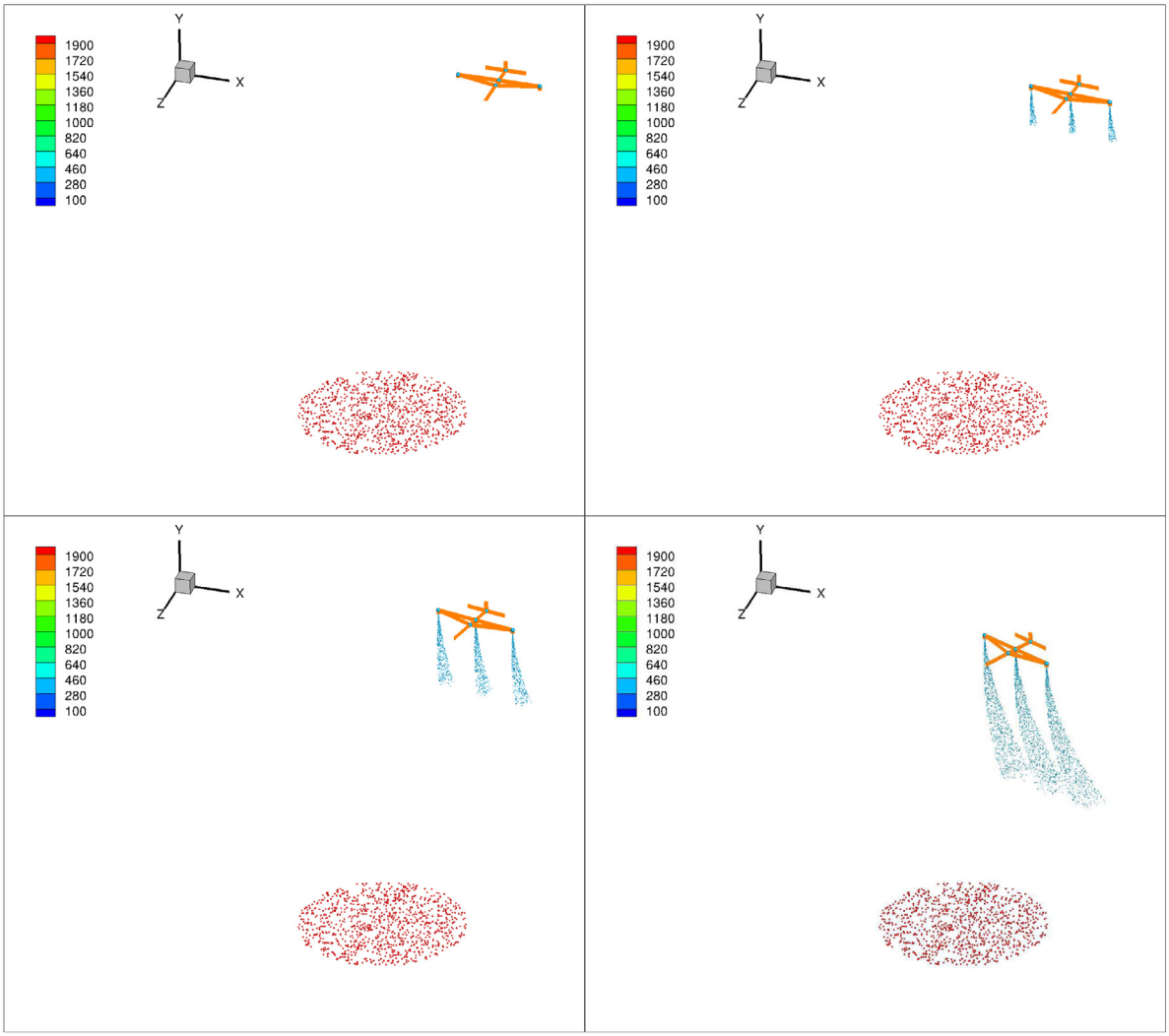

Fig. 4. Aerial simulation frames with 1000 targets, using the listed parameter set. In this case, the updraft was too strong with this mode of attack to be effective. This motivates the next section of Machine-Learning to seek more effective strategies.

- The release velocity, $\boldsymbol{v}(t=0) = 30$ m/s at a spray direction given by

$$\boldsymbol{n}^{spray} = \frac{\boldsymbol{n}^{plane-nor} + A\boldsymbol{n}^{rand}}{\|\boldsymbol{n}^{plane-nor} + A\boldsymbol{n}^{rand}\|}, \tag{5.7}$$

  where $\boldsymbol{n}^{plane-nor}$ is the unit normal to the plane fuselage, A is a spray cone amplitude and $\boldsymbol{n}^{rand}$ is a random unit vector.
- Density of retardants , $\rho_i = 1000$, kg/m^3,
- The size of the particles, $R_i = 0.1$ m,
- The drop rate of the particles, $\zeta = 30$, m^3/s and
- Density of air, $\rho_a = 1.225$, kg/m^3.

There is a designated updraft in the zone above the fire, with an upward velocity of warm air that decays from the source exponentially upwards, given by:

$$v^{up} = v_o^{up} e^{-ad_z}, \tag{5.8}$$

where d_z is the distance above ground and $0 \leq a$ is an updraft decay parameter (updrafts being highest at the source) Figs. 4–5 illustrate the action of the air-bomber and the resulting hits on the target. Approximately 19% of

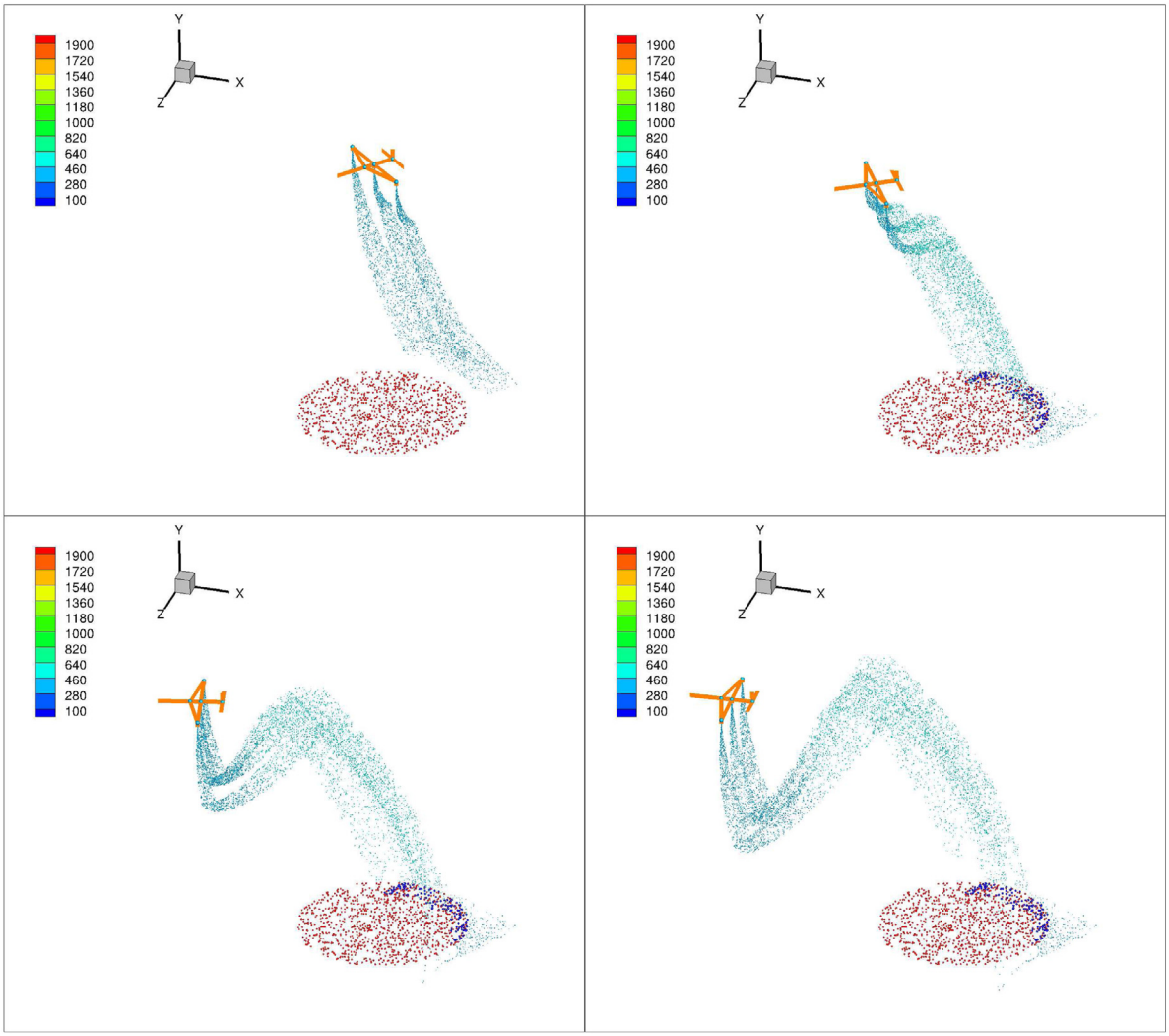

Fig. 5. Continuation of the frames in Fig. 4 for aerial simulation frames with 1000 targets, using the listed parameter set. In this case, the updraft was too strong with this mode of attack to be effective. This motivates the next section of Machine-Learning to seek more effective strategies.

the targets were hit. A "hit" was computed as follows:

$$\|\boldsymbol{r}_i - \boldsymbol{T}_j\| \leq \mathcal{F}(R_i, R_j^t), \tag{5.9}$$

where

$$\mathcal{F}(R_i, R_j^t) = \beta R_i + R_j^t, \tag{5.10}$$

where β_i represents the "splatter" of the retardant (for example $\beta = 10$) and R_j^t is the radius of the target. 1000 targets were used and an updraft decay parameter of $a = 0.001$. It is important to note that, in this case, the updraft was too strong with this mode of attack to be effective. This motivates the next section of Machine-Learning for more effective actions.

6. Machine-learning for system optimization

The rapid rate at which these simulations can be completed enables the ability to explore inverse problems seeking to determine what parameter combinations can deliver a desired result (Figs. 6). In order to cast the objective

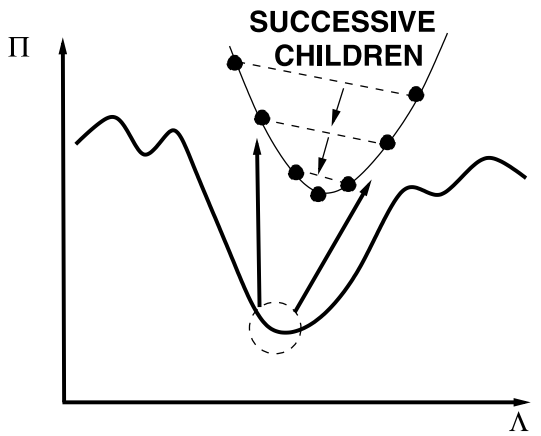

Fig. 6. The basic action of a MLA/GA-Machine Learning Algorithm/Genetic Algorithm. Zohdi [19–21,17].

mathematically, we set the problem up as a Machine Learning Algorithm (MLA); specifically a Genetic Algorithm (GA) variant, which is well-suited for nonconvex optimization. Following Zohdi [42–44,17] , we formulate the objective as a cost function minimization problem that seeks system parameters that match a desired response

$$\Pi^{tot}(\lambda_1, \ldots \lambda_{13}) = w_1 \Pi^{(1)} + w_2 \Pi^{(2)} \tag{6.1}$$

where

$$\Pi^{(1)} = (1 - \frac{H^h}{H^{tot}}) \qquad \text{and} \qquad \Pi^{(2)} = (1 - \frac{P^h}{P^{tot}}), \tag{6.2}$$

where H^{tot} is the total number of targets and H^h is the total number of targets hit and where P^{tot} is the total number of particles released and P^h is the total number of particles that hit the targets. We systematically minimize Eq. (6.1), $min_{\Lambda} \Pi$, by varying the design parameters: $\boldsymbol{\Lambda}^i \overset{\text{def}}{=} \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \ldots, \Lambda_N^i\} \overset{\text{def}}{=} \{flow\, rate, aircraft\, dynamics \ldots\}$. The system parameter search is conducted within the constrained ranges of $\Lambda_1^{(-)} \leq \Lambda_1 \leq \Lambda_1^{(+)}$, $\Lambda_2^{(-)} \leq \Lambda_2 \leq \Lambda_2^{(+)}$ and $\Lambda_3^{(-)} \leq \Lambda_3 \leq \Lambda_3^{(+)}$, etc. These upper and lower limits would, in general, be dictated by what is physically feasible.

6.1. System parameter search: Machine Learning Algorithm (MLA)

Here we follow Zohdi [42–44,17] in order to minimize Eq. (6.1), which we will refer to as a "cost function". Cost functions such as Eq. (6.1) are nonconvex in design parameter space and often nonsmooth. Their minimization is usually difficult with direct application of gradient methods. This motivates nonderivative search methods, for example those found in Machine Learning Algorithms (MLA's). One of the most basic subsets of MLA's are so-called Genetic Algorithms (GA's). Typically, one will use a GA first in order to isolate multiple local minima, and then use a gradient-based algorithm in these locally convex regions or reset the GA to concentrate its search over these more constrained regions. GA's are typically the simplest scheme to start the analysis, and one can, of course, use more sophisticated methods if warranted. For a review of GA's, see the pioneering work of John Holland [45,46], as well as Goldberg [47], Davis [48], Onwubiko [49] and Goldberg and Deb [50].

6.1.1. Generalities

The MLA/GA approach is extremely well-suited for nonconvex, nonsmooth, multicomponent, multistage systems and, broadly speaking, involves the following essential concepts:

1. **POPULATION GENERATION:** Generate a parameter population of genetic strings: $\boldsymbol{\Lambda}^i$
2. **PERFORMANCE EVALUATION:** Compute performance of each genetic string: $\Pi(\boldsymbol{\Lambda}^i)$
3. **RANK STRINGS:** Rank them $\boldsymbol{\Lambda}^i$, $i = 1, \ldots, N$
4. **MATING PROCESS:** Mate pairs/produce offspring

5. **GENE ELIMINATION:** Eliminate poorly performing genetic strings
6. **POPULATION REGENERATION:** Repeat process with updated gene pool and new *random* genetic strings
7. **SOLUTION POST-PROCESSING:** Employ gradient-based methods afterwards in local "valleys"-*if smooth enough*

6.1.2. Specifics

Following Zohdi [42–44,17], the algorithm is as follows:

- **STEP 1:** Randomly generate a population of S starting genetic strings, $\boldsymbol{\Lambda}^i, (i = 1, 2, 3, \ldots, S) : \boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \Lambda_4^i, \ldots, \Lambda_N^i\}$
- **STEP 2:** Compute fitness of each string $\Pi(\boldsymbol{\Lambda}^i)$, (i = 1, . . . , S)
- **STEP 3:** Rank genetic strings: $\boldsymbol{\Lambda}^i$, (i = 1, . . . , S)
- **STEP 4:** Mate nearest pairs and produce two offspring, (i = 1, . . . , S):
  $\boldsymbol{\lambda}^i \stackrel{\text{def}}{=} \Phi^{(I)} \circ \boldsymbol{\Lambda}^i + (1 - \Phi^{(I)}) \circ \boldsymbol{\Lambda}^{i+1}$, and $\boldsymbol{\lambda}^{i+1} \stackrel{\text{def}}{=} \Phi^{(II)} \circ \boldsymbol{\Lambda}^i + (1 - \Phi^{(II)}) \circ \boldsymbol{\Lambda}^{i+1}$,
  where for this operation, $\Phi^{(I)}$ and $\Phi^{(II)}$ are random numbers, such that $0 \leq \Phi^{(I)} \leq 1$, $0 \leq \Phi^{(II)} \leq 1$, which are different for each component of each genetic string
- **STEP 5:** Eliminate the bottom $M < S$ strings and keep top $K < N$ parents and top K offspring (K offspring+K parents+$M = S$)
- **STEP 6:** Repeat STEPS 1–6 with top gene pool (K offspring and K parents), plus M new, randomly generated, strings
- **IMPORTANT OPTION:** Rescale and restart search around best performing parameter set every few generations

Remark 4. If one selects the mating parameter Φ to be greater than one and/or less than zero, one can induce "mutations", i.e. characteristics that neither parent possesses. However, this is somewhat redundant with introduction of new random members of the population in the current algorithm.

Remark 5. If one does not retain the parents in the algorithm above, it is possible that inferior performing offspring may replace superior parents. Thus, top parents should be kept for the next generation. This guarantees a monotone reduction in the cost function. Furthermore, retained parents do not need to be re-evaluated-making the algorithm less computationally less expensive, since these parameter sets do not have to be reevaluated (or ranked) in the next generation. Numerous studies of the author have shown that the advantages of parent retention outweighs inbreeding, for sufficiently large population sizes. Finally, we remark that this algorithm is easily parallelizable.

Remark 6. After application of such a global search algorithm, one can apply a gradient-based method, if the objective function is sufficiently smooth in that region of the parameter space. In other words, if one has located a convex portion of the parameter space with a global genetic search, one can employ gradient-based procedures locally to minimize the objective function further, since they are generally much more efficient for convex optimization of smooth functions. An exhaustive review of these methods can be found in the texts of Luenberger [51] and Gill, Murray and Wright [52].

6.2. Model problem-parameter settings

We applied the MLA algorithm, with the following parameters (thirteen):

1. $\boldsymbol{\Lambda}_1$ =the plane's velocity (magnitude): v^{plane},
2. $\boldsymbol{\Lambda}_2, \boldsymbol{\Lambda}_3, \boldsymbol{\Lambda}_4$ = the plane's angular velocity: $\boldsymbol{\omega}^{plane} = (\omega_x, \omega_y, \omega_z)$,
3. $\boldsymbol{\Lambda}_5, \boldsymbol{\Lambda}_6, \boldsymbol{\Lambda}_7$ =the initial plane position, $\boldsymbol{r}(0) = (r_x(0), r_y(0), r_z(0))$,
4. $\boldsymbol{\Lambda}_8$ = the particle size, R_i,
5. $\boldsymbol{\Lambda}_9$ = the sprayer amplitude A,
6. $\boldsymbol{\Lambda}_{10}$ = the release time start, T_I,
7. $\boldsymbol{\Lambda}_{11}$ = the release time end, T_F,

Table 1
The top system parameter performers ($\Lambda_1 - \Lambda_{13}$).

Λ_1	Λ_2	Λ_3	Λ_4	Λ_5	Λ_6	Λ_7	Λ_8	Λ_9	Λ_{10}	Λ_{11}	Λ_{12}	Λ_{13}	Π
217.194	1.894	0.251	−0.952	43.473	129.806	−39.9115	0.124	0.337	0.155	0.936	27.736	309.963	0.146

8. Λ_{12} = the drop rate of retardant: ζ and
9. Λ_{13} = the particle drop velocity, v^{drop}.

Explicitly, the design vector is:

$$\Lambda = \{v^{plane}, \omega_x, \omega_y, \omega_z, r_x(0), r_y(0), r_z(0), R_i, A, T_i, T_e, \zeta, v^{drop}\}. \tag{6.3}$$

The initial search range:

1. 100 km/h = $100 \times 1000/3600$ m/s = $\Lambda_1^- \leq \Lambda_1^- \leq \Lambda_1^+ = 1000$ km/h = $1000 \times 1000/3600$ m/s,
2. $(-2, -2, -2)$ rad/s = $(\Lambda_2^-, \Lambda_3^-, \Lambda_4^-) \leq (\Lambda_2, \Lambda_3, \Lambda_4) \leq (\Lambda_2^+, \Lambda_3^+, \Lambda_4^+) = (2, 2, 2)$ rad/s,
3. $(-100, 100, -100)$ m = $(\Lambda_5^-, \Lambda_6^-, \Lambda_7^-) \leq (\Lambda_5, \Lambda_6, \Lambda_7) \leq (\Lambda_5^+, \Lambda_6^+, \Lambda_7^+) = (100, 250, 100)$ m,
4. 0.05 m$\Lambda_8^- \leq \Lambda_8 \leq \Lambda_8^+ = 0.2$ m,
5. $0.0 = \Lambda_9^- \leq \Lambda_9 \leq \Lambda_9^+ = 0.5$,
6. $0.0T = \Lambda_{10}^- \leq \Lambda_{10} \leq \Lambda_{10}^+ = 0.25T$,
7. $0.25T = \Lambda_{11}^- \leq \Lambda_{11} \leq \Lambda_{11}^+ = T$,
8. 10 m^3/s = $\Lambda_{12}^- \leq \Lambda_{12} \leq \Lambda_{12}^+ = 60$ m^3/2,
9. 100 m/s = $\Lambda_{13}^- \leq \Lambda_{13} \leq \Lambda_{13}^+ = 400$ m/s,

which of course will change if the search re-adaptation option is chosen. Additionally, we used

- Time: $T = 2.5$ s and
- Design weights: w_1 and $w_2 = 1$.

6.2.1. Numerical results

Fig. 7 shows the reduction of the cost function for the 13 parameter set. This cost function Π represents the percentage of unburned material left in the zone of interest. In other words, the system is being driven to the parameters generating the worst case scenario. Shown are the best performing gene (design parameter set, in *red*) as a function of successive generations, as well as the average performance of the entire population of the genes (designs, in *green*). The design parameters $\Lambda = \{\Lambda_1, \Lambda_2 \dots \Lambda_N\}$ are optimized over the search intervals (13 variables): $\Lambda_i^- \leq \Lambda_i \leq \Lambda_i^+$, $i = 1, 2, \dots, 13$. We used the following MLA settings:

- Number of design variables: 13,
- Population size per generation: 50,
- Number of parents to keep in each generation: 10,
- Number of children created in each generation: 10,
- Number of completely new genes created in each generation: 30
- Number of generations for re-adaptation around a new search interval: 20 and
- Number of generations: 400.

The algorithm was automatically reset every *20 generations*. The entire 400 generation simulation, with 50 genes per evaluation (20000 total designs), took a few minutes on a laptop, *making it ideal as a design tool.* Fig. 7 (average population of 50 genes performance and top gene performance) and Table 1 (values of the gene components) illustrate the results. Note that the MLA/GA readapts every 20 generations, leading to the nonmonotone reduction of the cost function. Often, this action is more efficient than allowing the algorithm to not readapt, since it probes around the current optimum for better alternatives. This allows system designers more flexibility in parameter selection. We note that, for a given set of parameters, a complete simulation takes on the order of 0.025 s, thus over 100,000 parameter sets can be evaluated in an hour, *without even exploiting the inherent parallelism of the MLA/GA.*

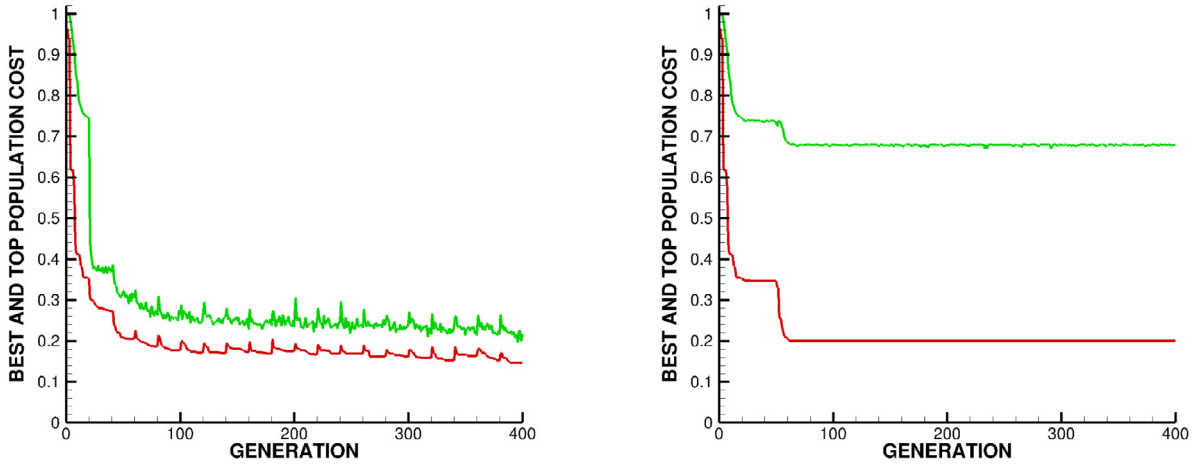

Fig. 7. The reduction of the cost function for the 13 parameter set. Shown are the best performing gene (design parameter set, in *red*) as a function of successive generations, as well as the average performance of the entire population of genes (designs, in *green*). Left: Allowing the MLA/GA to readapt every 20 generations, leading to the nonmonotone reduction of the cost function. Often, this action is more efficient than allowing the algorithm not readapt, since it probes around the current optimum for better local alternatives—*as is the case in this example*. Right: With no readaptation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7. Conclusions and practical usage

In closing, the objective of this work was to develop a computational framework for aerial drops of fire retardants in dangerous fire environments by developing a meshless discrete element model that tracks the trajectory of released airborne materials from a controlled aircraft, ranging from retardant powders to encapsulated packets, subjected to prevailing wind velocities and fire-driven updrafts. A Machine Learning Algorithm (MLA) was then used to rapidly ascertain the optimal aircraft (unmanned or manned) dynamics to maximize the fire-retardant release effectiveness (released material usage and target impact). The framework was designed to enable rapidly computable Digital Twin type technologies, i.e. digital replicas that run in real time with the physical system. However, it was also designed to run at much faster rates, in order to enable MLA's to optimize the planning, by running quickly on laptops and mobile systems. The overall guiding motivation is to provide a useful tool to enable rapid flight-path planning for aerial first-responders *in real-time* and to train pilots. Numerical examples were provided to illustrate the process. The approach can be easily combined with meshless fire propagation models, such as those illustrated in Zohdi [17] or other paradigms such as $FARSITE$ (https://www.firelab.org/project/farsite), which was developed over several years using empirical data to characterize fire behavior over 2D terrain and is based on Rothermel's spread model and subsequent extensions (see Finney [53] and Andrews [54] for comprehensive reviews). The empirical basis for such a model is somewhat limited and its utility for fire scenarios depends heavily on the expertise of the fire behavior analyst operating the model. Oftentimes, fuel and wind speed factors are adjusted in real time to improve fire behavior predictions—*this task is ideal for MLA's* and the framework developed in this work. More recently, physics-based fire models like FIRETEC (https://www.frames.gov/firetec/home) have been developed at Los Alamos National Laboratory (LANL) by coupling physics-based fire/atmosphere models and high fidelity computational fluid dynamics (CFD) to simulate 3D fire/wind/fuel/terrain interaction. However, running FIRETEC requires the use of significant computing power such as those available at national laboratories. While FIRETEC has proven to be a valuable tool for studying wildfire behavior and gaining understanding of the ways that physical processes connect wildfires to their environment, it is too computationally expensive for real time use. For general conditions, there can be cases where the change in the surrounding fluid's behavior, due to the motion of the aircraft, released particles etc., may be important. The result is a system of coupled equations between the aircraft, particles, fire and the environment, requiring spatio-temporal discretization, using high-fidelity Finite Element or Finite Difference methods, of the classical equations governing the surrounding fluid mechanics (Navier–Stokes, see the Appendix).[3]

[3] Other computationally-oriented codes include, the Fire Dynamics Simulation (FDS, from NIST) https://pages.nist.gov/fds-smv/ and WRF-FIRE (from NCAR) https://www.openwfm.org/wiki/WRF-Fire, etc.

Generally such models are ineffective for rapid real-time use, but are quite useful for detailed offline background analyses, where a rapid response is a nonissue. In those paradigms, for example, for the released material, the continuum fluid discretization is usually combined with a Discrete Element Method for the particle dynamics. We refer the reader to Avci and Wriggers [55], Onate et al. [56,57], Leonardi et al. [58], Onate et al. [59], Bolintineanu et al. [60] and Zohdi [22,25]. Such models are significantly more complex than the models used in the current work. However, while useful in many industrial applications where high precision is required, the use of such detailed approaches is unwarranted for the present work. Other approaches include the QUIC-FIRE (Goodrick et al. [61]) paradigm, which was developed at LANL to allow faster turnaround of fire modeling using a cellular automata-based approach to approximate the CFD calculations. In addition to real time decision support, QUIC-FIRE was developed with the intention of enabling ensemble-simulation-based forecasts to capture the probabilistic nature of fire events, *ultimately employing MLAs*. Regardless of the model used, the overall goal must be to provide accurate and real-time feedback to deployed firefighters by incorporating rapid data collection from satellites, UAVs, social media, etc. Thus, any modern simulation paradigm must be able to adapt to rapid changes in the environment, autonomously. Autonomous capabilities are critical for rapid operations large distances, resulting in communication delays and the lack of 24/7 connectivity (i.e. limited ground stations, telemetry, command networks, etc.). These scenarios make demands that are beyond human reaction/decision times. In this vein, the use of multiple UAVs may be useful, and we refer the reader to Zohdi [44,17] for more details.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. High-fidelity models

For general conditions, there can be cases where the change in the surrounding fluid's behavior, due to the motion of the aircraft, released particles, etc., may be important. High-fidelity models would require discretization of the Navier–Stokes equations:

$$\begin{aligned} &Balance\ of\ mass: \frac{\partial \rho}{\partial t} = -\nabla_x \rho \cdot \boldsymbol{v} - \rho \nabla_x \cdot \boldsymbol{v}, \\ &Balance\ of\ momentum: \rho(\frac{\partial \boldsymbol{v}}{\partial t} + (\nabla_x \boldsymbol{v}) \cdot \boldsymbol{v}) = \nabla_x \cdot \boldsymbol{\sigma} + \boldsymbol{f}, \\ &Constitutive\ Law: \boldsymbol{\sigma} = -P\mathbf{1} + \lambda tr\boldsymbol{D}\mathbf{1} + 2\mu\boldsymbol{D} = -P\mathbf{1} + 3\kappa\frac{tr\boldsymbol{D}}{3}\mathbf{1} + 2\mu\boldsymbol{D}', \end{aligned} \tag{A.1}$$

where $\rho(\boldsymbol{x})$ is the density field of the fluid, $\boldsymbol{v}(\boldsymbol{x})$ is the fluid velocity field, $\boldsymbol{\sigma}(\boldsymbol{x})$ is the fluid stress field, $\boldsymbol{D}(\boldsymbol{x})$ is the fluid velocity gradient field, $\boldsymbol{f}(\boldsymbol{x})$ is the body force field, $P(\boldsymbol{x})$ is the fluid pressure field, $\lambda(\boldsymbol{x})$, $\kappa(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ are fluid material property fields.[4] It is important to emphasize that physically compatible boundary data must be applied, and this is not a trivial matter for compressible flow. Additionally, the first law of thermodynamics should be included (along with equations for various chemical reactions), which reads as

$$\rho\dot{w} - \boldsymbol{\sigma} : \nabla_x \boldsymbol{v} + \nabla_x \cdot \boldsymbol{q} - \rho z = 0, \tag{A.2}$$

where $w(x)$ is the stored energy in the fluid, $\boldsymbol{q}(\boldsymbol{x})$ is the heat flux field, z is the heat source field per unit mass. Such models are significantly more complex than the models used in the current work. More detailed analyses of fluid–particle interaction can be achieved in a direct, brute-force, numerical schemes, treating the particles as part of the fluid continuum (as another fluid or solid phase), and thus meshing them in a detailed manner. In such an approach (for example see Avci and Wriggers [55])

- A fluid-only problem is solved, with (instantaneous) boundary conditions of $\boldsymbol{v}^f(\boldsymbol{x}) = \boldsymbol{v}_i(\boldsymbol{x})$ at each point on the fluid–particle boundaries, where the velocity of the points on the boundary is given by

$$\boldsymbol{v}_i(\boldsymbol{x}) = \boldsymbol{v}_i^{cm} + \boldsymbol{\omega}_i \times \boldsymbol{R}_{cm\rightarrow surf.}(\boldsymbol{x}), \tag{A.3}$$

  where $\boldsymbol{v}_i^{cm}$ is the center of mass and $\boldsymbol{\omega}_i$ is the angular velocity for each of the individual particles and $\boldsymbol{R}_{cm\rightarrow surf.}$ is a vector from the mass center to the surface.

[4] It is customary to specify $\boldsymbol{v}$ and P on the boundary, and to determine ρ on the boundary through the Equation of State. P is given by an Equation of State.

- For each particle, one would solve:

$$m_i \dot{\boldsymbol{v}}_i = \boldsymbol{\Psi}_i^{drag} + other \ \ forces \tag{A.4}$$

and

$$I_i \dot{\boldsymbol{\omega}}_i = \boldsymbol{M}_i^{drag} + other \ \ moments \tag{A.5}$$

where the forces and moments would have a contribution from the fluid drag (with particle occupying domain Ω_i and outward surface normal $\boldsymbol{n}$) is defined as

$$\boldsymbol{\Psi}_i^{drag} = \int_{\partial \Omega_i} \boldsymbol{\sigma} \cdot \boldsymbol{n} \, dA, \tag{A.6}$$

and

$$\boldsymbol{M}_i^{drag} = \int_{\partial \Omega_i} \boldsymbol{R}_{cm \rightarrow surf.} \times \boldsymbol{\sigma} \cdot \boldsymbol{n} \, dA, \tag{A.7}$$

- At a time-step, the process is iteratively driven by solving the fluid-only problem first, then the particles-only problem, and repeated until convergence in an appropriate norm.

Along these lines, in Zohdi [22,25], more detailed, computationally intensive models were developed to characterize the motion of small-scale particles embedded in a flowing fluid where the dynamics of the particles affects the dynamics of the fluid and vice-versa. In such a framework, a fully implicit Finite-Difference discretization of the Navier–Stokes equations was used for the fluid and a direct particle-dynamics discretization is performed for the particles. Because of the large computational difficulty and expense of a conforming spatial discretization needed for large numbers of embedded particles, simplifying assumptions are made for the coupling, based on semi-analytical computation of drag-coefficients, which allows for the use of coarser meshes. Even after these simplifications, the particle–fluid system was strongly-coupled. The approach taken in that work was to construct a sub-model for each primary physical process. In order to resolve the coupling, a recursive staggering scheme was constructed, which was built on works found in Zohdi [22–25]. The procedure was as follows (at a given time increment): (1) each submodel equation (fluid or particle-system) is solved individually, "freezing" the other (coupled) fields in the system, allowing only the primary field to be active, (2) after the solution of each submodel, the associated field variable was updated, and the next submodel was solved and (3) the process is then repeated, until convergence. The time-steps were adjusted to control the rates of convergence, which is dictated by changes in the overall physics. Specifically, the approach was a staggered implicit time-stepping scheme, with an internal recursion that automatically adapted the time-step sizes to control the rates of convergence within a time-step. If the process did not converge (below an error tolerance) within a preset number of iterations, the time-step was adapted (reduced) by utilizing an estimate of the spectral radius of the coupled system. The developed approach can be incorporated within any standard computational fluid mechanics code based on finite difference, finite element, finite volume or discrete/particle element discretization (see Labra and Onate [62], Onate et al. [56,57], Rojek et al. [63] and Avci and Wriggers [55]). However, while useful in many industrial applications where high precision is required, the use of such models for the coarser applications of interest in this work is probably unwarranted in most cases.

References

[1] USDA Forest Service, Long-term retardant specifications, 2008, https://www.fs.fed.us/rm/fire/wfcs/ret.htm.

[2] USDA Forest Service, Aerial firefighting use and effectiveness (AFUE): Preliminary findings, 2019, https://www.fs.usda.gov/sites/default/files/2019-07/2019_03_15_19_bp_afue_2019_update_final_web.pdf.

[3] Wildfire fighting: Provincial and territorial approaches to air tankers. Canadian American strategic review, 2016, Archived from the original on 2016-06-11.

[4] Bushfire Cooperative Research Center, Effectiveness and efficiency pf aerial fire fighting in Australia, Fire Note (50) (2009) 1–4.

[5] Association of aerial firefighters, http://airtanker.org/.

[6] Aerial fire fighting, Wikipedia, https://en.wikipedia.org/wiki/Aerial_firefighting.

[7] B. Christopher, Does using airplanes to put out forest fires actually work? 2016, https://priceonomics.com/does-using-airplanes-to-put-out-forest-fires/.

[8] B. Gabbert, Like a phoenix, the supertanker to rise again, Five Aviation, 2015, https://fireaviation.com/2015/08/12/like-the-phoenix-the-supertanker-to-rise-again/.

[9] W. Jendsch, Aerial Firefighting, Schiffer Publishing, ISBN: 978-0-7643-3068-1, 2007.
[10] National Transportation Safety Board Aviation Accident Final Report, Report number: LAX95GA219B, 1995.
[11] National Transportation Safety Board Aviation Accident Final Report, Report number: LAX94FA323, 1996.
[12] C. Sanford, Water bombs- Boeing invention could make C-17 a firefighter, Boeing Front. 2 (4) (2003) https://www.boeing.com/news/frontiers/archive/2003/august/i_ids4.html.
[13] Boeing Patent. Firefighting bomblets and a precision aerial firefighting method utilizing the same Abstract. https://patents.google.com/patent/US20060011355A1/en.US20060011355A1.
[14] Top 10 best fire estinguishing balls in 2020, 2019, https://superiortoplist.com/best-fire-extinguisher-balls/.
[15] E. Limer, Just chuck this ball at a fire to put it out with an explosion- the elide fire ball doesn't mess around, 2016, https://www.popularmechanics.com/technology/gadgets/a19862/elide-fire-ball-throwable-fire-extinguisher/.
[16] L. Chang, Extinguish a fire by throwing this ball into it, Digital Trends, 2016, https://www.digitaltrends.com/cool-tech/elide-fire-ball/.
[17] T.I. Zohdi, A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments, Comput. Methods Appl. Mech. Engrg. (2020) http://dx.doi.org/10.1016/j.cma.2020.112907.
[18] P. Schulthess, M. Neuenschwander, K.M. Mosalam, M. Knobloch, A computationally rigorous approach to hybrid fire testing, Comput. Struct. 238 (1) (2020) 106301.
[19] T.I. Zohdi, A note on firework blasts and qualitative parameter dependency, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. (2016) http://dx.doi.org/10.1098/rspa.2015.0720.
[20] T.I. Zohdi, J. Cabalo, On the thermomechanics and footprint of fragmenting blasts, Internat. J. Engrg. Sci. 118 (2017) 28–39.
[21] T.I. Zohdi, Modeling the spatio-thermal fire hazard distribution of incandescent material ejecta in manufacturing, Comput. Mech. (2018) http://dx.doi.org/10.1007/s00466-018-1617-2.
[22] T. Zohdi, Computation of strongly coupled multifield interaction in particle-fluid systems, Comput. Methods Appl. Mech. Engrg. 196 (2007) 3927–3950.
[23] T.I. Zohdi, On the dynamics of charged electromagnetic particulate jets, Arch. Comput. Methods Eng. 17 (2) (2010) 109–135.
[24] T.I. Zohdi, Numerical simulation of charged particulate cluster-droplet impact on electrified surfaces, J. Comput. Phys. 233 (2013) 509–526.
[25] T.I. Zohdi, Embedded electromagnetically sensitive particle motion in functionalized fluids, Comput. Part. Mech. 1 (2014) 27–45.
[26] G. Plimpton, Fireworks: A History and Celebration, Doubleday, ISBN: 0-385-15414-3, 1984.
[27] A. St. Hill Brock, A History of Fireworks, George G. Harrap and Co., 1949.
[28] M.S. Russell, The Chemistry of Fireworks, Royal Society of Chemistry, Great Britain, ISBN: 978-0-85404-127-5, 2008.
[29] T. Shimizu, Fireworks: The Art, Science, and Technique, Pyrotechnica Publications, ISBN: 978-0-929388-05-2, 1996.
[30] S. Werrett, Fireworks: Pyrotechnic Arts and Sciences in European History, University of Chicago Press, ISBN: 978-0-226-89377-8, 2010.
[31] S. Kazuma, Hanabi No Hon. Fireworks Book, Tankosha, ISBN: 4-473-03177-2, 2004.
[32] S. Kazuma, Wonder of Fireworks, Soft Bank Creative, ISBN: 978-4-7973-6450-7, 2011.
[33] V.K. Wingerden, I. Hesby, R. Eckhoff, Ignition of dust layers by mechanical sparks, in: Proceedings of 7th Global Congress on Process Safety, Chicago, Ill, 2011.
[34] A.C. Fernandez-Pello, Wildland fire spot ignition by sparks and firebrands, Fire Saf. J. 91 (2017) 2–10.
[35] G.E. Pleasance, J.A. Hart, An Examination of Particles from Conductors Clashing as Possible Source of Bushfire Ignition, Report FM-1:1977, State Electricity Commission of Victoria (SEC), Research and Development Department, Victoria, Australia, 1977.
[36] A.D. Stokes, Fire ignition by copper particles of controlled size, J. Electr. Electron. Eng. Aust. 10 (1990) 188–194.
[37] G. Rowntree, A. Stokes, Fire ignition of aluminum particles of controlled size, J. Electr. Electron. Eng. 11 (1994) 7–123.
[38] R. Hadden, S. Scott, C. Lautenberger, C.A. Fernandez-Pello, Ignition of combustible fuel beds by hot particles: an experimental and theoretical study, Fire Technol. 47 (2011) 341–355.
[39] J.L. Urban, C.D. Zak, J. Song, A.C. Fernandez-Pello, Smolder spot ignition of natural fuels by a hot metal particle, Proc. Combust. Inst. (ISSN: 1540-7489) 36 (2) (2017) 3211–3218, http://dx.doi.org/10.1016/j.proci.2016.09.014.
[40] C.Y. Chow, An Introduction to Computational Fluid Dynamics, Wiley, New York, 1980.
[41] H. Schlichtling, Boundary-Layer Theory, seventh ed., McGraw-Hill, New York, 1979.
[42] T.I. Zohdi, Mechanistic modeling of swarms, Comput. Methods Appl. Mech. Engrg. 198 (21–26) (2009) 2039–2051.
[43] T.I. Zohdi, Multiple UAVs for mapping: a review of basic modeling, simulation and applications, Annu. Rev. Environ. Resour. (2018) http://dx.doi.org/10.1146/annurev-environ-102017-025912.
[44] T.I. Zohdi, The game of drones: rapid agent-based machine-learning models for multi-UAV path planning, Comput. Mech. (2019) http://dx.doi.org/10.1007/s00466-019-01761-9.
[45] J.H. Holland, Adaptation in Natural & Artificial Systems, University of Michigan Press, Ann Arbor, Mich, 1975.
[46] J.H. Holland, J.H. Miller, Artificial adaptive agents in economic theory (PDF), Amer. Econ. Rev. 81 (2) (1991) 365–371, Archived from the original (PDF) on October 27, 2005.
[47] D.E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, 1989.
[48] L. Davis, Handbook of Genetic Algorithms, Thompson Computer Press, 1991.
[49] C. Onwubiko, Introduction to Engineering Design Optimization, Prentice Hall, 2000.
[50] D.E. Goldberg, K. Deb, Special issue on genetic algorithms, Comput. Methods Appl. Mech. Engrg. 186 (2–4) (2000) 121–124.
[51] D. Luenberger, Introduction to Linear & Nonlinear Programming, Addison-Wesley, Menlo Park, 1974.
[52] W. Gill, M. Wright, Practical Optimization, Academic Press, 1995.
[53] M.A. Finney, FARSITE: Fire Area Simulator-Model Development and Evaluation, Research Paper, RMRS-RP-4-Revised, United States Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2004.

[54] P.L. Andrews, The Rothermel Surface Fire Spread Model and Associated Developments: A Comprehensive Explanation, General Technical Report, RMRS-GTR-371, United States Department of Agriculture, Forest Service, Rocky Mountain Research Station, Fort Collins, CO, 2018.
[55] B. Avci, P. Wriggers, A DEM-FEM coupling approach for the direct numerical simulation of 3D particulate flows, J. Appl. Mech. 79 (2012) 010901-(1-7).
[56] E. Onate, S.R. Idelsohn, M.A. Celigueta, R. Rossi, Advances in the particle finite element method for the analysis of fluid-multibody interaction and bed erosion in free surface flows, Comput. Methods Appl. Mech. Engrg. 197 (19–20) (2008) 1777–1800.
[57] E. Onate, M.A. Celigueta, S.R. Idelsohn, F. Salazar, B. Suarez, Possibilities of the particle finite element method for fluid-soil-structure interaction problems, Comput. Mech. 48 (2011) 307–318.
[58] A. Leonardi, F.K. Wittel, Mendoza M, H.J. Herrmann, Coupled DEM-LBM method for the free-surface simulation of heterogeneous suspensions, Comput. Part. Mech. 1 (1) (2014) 3–13.
[59] E. Onate, M.A. Celigueta, S. Latorre, G. Casas, R. Rossi, J. Rojek, Lagrangian analysis of multiscale particulate flows with the particle finite element method, Comput. Part. Mech. 1 (1) (2014) 85–102.
[60] D.S. Bolintineanu, G.S. Grest, J.B. Lechman, F. Pierce, S.J. Plimpton, P.R. Schunk, Particle dynamics modeling methods for colloid suspensions, Comput. Part. Mech. 1 (3) (2014) 321–356.
[61] S. Goodrick, R. Linn, S. Brambilla, Comparing HIGRAD/FIRETEC to QUIC-Fire, a coupled fire-atmosphere model for operational applications, 2018, https://ui.adsabs.harvard.edu/abs/2018AGUFMEP31B.07G.
[62] C. Labra, E. Onate, High-density sphere packing for discrete element method simulations, Commun. Numer. Methods. Eng. 25 (7) (2009) 837–849.
[63] J. Rojek, C. Labra, O. Su, E. Onate, Comparative study of different discrete element models and evaluation of equivalent micromechanical parameters, Int. J. Solids Struct. 49 (2012) 1497–1517, http://dx.doi.org/10.1016/j.ijsolstr.2012.02.032.