ORIGINAL PAPER



An agent-based computational framework for simulation of global pandemic and social response on *planet X*

T. I. Zohdi¹

Received: 24 June 2020 / Accepted: 12 July 2020 © Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

The increase in readily available computational power raises the possibility that direct agent-based modeling can play a key role in the analysis of epidemiological population dynamics. Specifically, the objective of this work is to develop a robust agent-based computational framework to investigate the emergent structure of Susceptible-Infected-Removed/Recovered (SIR)-type populations and variants thereof, on a global planetary scale. To accomplish this objective, we develop a planet-wide model based on interaction between discrete entities (agents), where each agent on the surface of the planet is initially uninfected. Infections are then seeded on the planet in localized regions. Contracting an infection depends on the characteristics of each agent—i.e. their susceptibility and contact with the seeded, infected agents. Agent mobility on the planet is dictated by social policies, for example such as "shelter in place", "complete lockdown", etc. The global population is then allowed to evolve according to infected states of agents, over many time periods, leading to an SIR population. The work illustrates the construction of the computational framework and the relatively straightforward application with direct, non-phenomenological, input data. Numerical examples are provided to illustrate the model construction and the results of such an approach.

Keywords Pandemic · Agent-based · Simulation

1 Introduction

The COVID-19 pandemic of 2020 has led to a significant increase in research in the area of modeling and simulation of infectious diseases. There are numerous aspects associated with this epoch-changing event that is now facing humanity. Macroscale (planetary) disease propagation, in addition to the related issues of logistical and political responses, is a central issue. Accordingly, the objective of this work is to develop a computationally-amenable agent-based model to investigate the behavior of an infected population by directly working at the individual-to-individual level of interaction. The wide-spread availability of computational power now raises the possibility that robust agent-based modeling can play a significant role in the analysis of infectious disease propagation. The key feature of agent-based modeling is

⊠ T. I. Zohdi zohdi@me.berkeley.edu that discrete entities (agents) are used to directly represent a population (Fig. 1). This enables the detailed analysis of epidemiological population dynamics and the ability to investigate the emergent structure SIR-type (Susceptible-Infected-Removed/Recovered) populations, as well more complex extensions, due to initially localized infections within a population on a global planetary scale, including the effects of social responses.

1.1 Classical basic models

Before proceeding with the construction of an agent-based approach, it is useful to review basic concepts in the analysis of population dynamics, which dates back over two centuries to the work of Thomas Malthus. In 1798, he postulated that a population, denoted p, at a future time $(t + \Delta t)$, is related to the current population (at time t) by

 $p(t + \Delta t) = \lambda p(t), \tag{1.1}$

¹ Department of Mechanical Engineering, University of California, 6195 Etcheverry Hall, Berkeley, CA 94720-1740, USA



Fig. 1 A model problem of a "planet" with a population, which experiences sudden localized infections. Left: a model schematic and right: a computational model (blue representing currently uninfected and green representing infected)

where $\lambda = 1 + (b - d)\Delta t$, where b is a birth rate parameter and where d is a death rate parameter. One may write

$$p(t + \Delta t) - p(t) = (\lambda - 1)p(t),$$
 (1.2)

leading to, in the limit as $\Delta t \rightarrow 0$,

$$\frac{\Delta p}{p} = (b-d)\Delta t \Rightarrow \frac{dp}{p} = (b-d)dt.$$
(1.3)

Integrating and applying the initial condition p(t = 0) = p(0) yields

$$ln(p(t)) = c + (b - d)t \Rightarrow p(t) = e^{c + (b - d)t} \Rightarrow p(t)$$
$$= p(0)e^{(b - d)t}.$$
(1.4)

Variants/extensions of this simple model include interacting subpopulations. One family of models are of particular interest in this work, namely the so-called SIR-type (Susceptible-Infected/Infectious-Removed/Recovered), described next. Subsets of the population are assigned either S, I, or R status. The genesis of such models is the paper from 1927 of Kermack and McKendrick [35].

1.2 SIR sub-population models

SIR models identify three subpopulation classes of individuals, with an assumption that the overall population is constant, since the pandemic/epidemic time scales are faster than birth time scales. The following are key:

- S "Susceptible", that can contract diseases,
- I = "Infected", that can transmit the disease and who are infected and

- R = "Recovered/Removed" (dead or immune), where the typical assumptions include: (a) the gain in the infected class is at a rate proportional to the number of infected and susceptible, that is k_1SI , $k_1 > 0$ (b) the rate of removal of the infected is proportional to the number of infected, k_2I , $k_2 > 0$ and (c) the incubation period is short enough to be negligible.
- In addition, if it is assumed that the various classes are uniformly mixed, where for the susceptible population:

$$\frac{dS}{dt} = -k_1 SI,\tag{1.5}$$

while for the infected subpopulation

$$\frac{dI}{dt} = k_1 S I - k_2 I, \tag{1.6}$$

and for the removed population

$$\frac{dR}{dt} = k_2 I. \tag{1.7}$$

· Adding all of the populations together yields

$$\frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = 0, \qquad (1.8)$$

where

$$S + I + R = p, \tag{1.9}$$

where *p* is the total population, $S(0) = S_o > 0$, $I(0) = I_o > 0$, R(0) = 0, $k_1 > 0$ is the infection rate and $k_2 > 0$ is the death rate.

A crucial question is, under what conditions does an infection grow, i.e. an "epidemic" occur? From Eq. 1.6, if

$$k_1 SI - k_2 I > 0 \Rightarrow k_1 SI > k_2 I \Rightarrow \frac{k_1}{k_2} S > 1, \qquad (1.10)$$

and thus

$$\frac{dI}{dt} > 0. \tag{1.11}$$

Equation 1.10 provides the threshold for the susceptible population

$$\frac{k_1}{k_2}S > 1 \Rightarrow S > \frac{k_2}{k_1} \tag{1.12}$$

to allow growth to occur. The parameter k_2/k_1 is sometimes called the infectious contact rate, while its reciprocal is called the recovery/removal rate. If we assume that everyone in the beginning is susceptible (S(t = 0) = 1), then Eq. 1.10 implies that

$$\frac{k_1}{k_2} > 1$$
 (1.13)

is the condition for growth. This simple model is one of the most basic to describe epidemics. For reviews, we refer the reader to Murray [42].

1.3 Generalization of the SIR family of models

The basic SIR model can be extended in the following ways:

- *The SIS model*: Susceptible-Infected-Susceptible (again), typified by the common cold.
- *The SIRD model*: Susceptible-Infected-Recovered-Deceased, which distinguished between recovered and dead.
- *The MSIR model*: Maternal-Susceptible-Infected-Recovered, where the "M class" stands for immunity derived from the mother.
- *The SEIR model*: Susceptible-Exposed-Infected-Recovered, which distinguishes between "infected" and "exposed".
- *The SEIS model*: Susceptible-Exposed-Infected-Susceptible (again), typified by the common cold, yet distinguishes between "infected" and "exposed".
- *The MSEIR model*: Maternal-Susceptible-Exposed-Infected-Recovered, which incorporates features of the models above.
- *The MSEIRS model*: Maternal-Susceptible-Exposed-Infected-Recovered-Susceptible, which incorporates all of the features of the models above.



Fig. 2 The classical process of developing a continuum model from an inherently discrete system, which is then re-discretized into nodes. Information is lost in this "homogenization" process (Zohdi [53])

For overviews and details on these various models, we refer the reader to Kermack and McKendrick [35], Murray [42], Hethcote[30,31], Harko et al. [29], Baily [3], Altizer and Nunn [1], Miller [40], Miller [41], Osemwinyen and Diakhaby [44], Brauer and Castillo [12] Anderson [2], Barlett [5], May and Anderson [38], Capasso [15] and Vynnycky and White [48]. Various features can be included, such as

- Variable contact rates,
- Adult vaccinations,
- Child vaccinations,
- Newborn vaccinations,
- Effects of age and
- Vector transmission (for example from mosquitos).

Virtually all subsequent, more complex spatio-temporal extensions construct homogenized continuum (PDE-based) models, incorporating the above models. These approaches require extensive, complex, discretization techniques and are of limited value for studies on population dynamics with underlying complex interaction between subpopulations (Fig. 2). Such models have limited predictive capability and are computationally expensive, due to the extremely fine discretization needed to achieve tolerable numerical accuracy. Independent of the numerical difficulties, such modeling approaches attempt to develop continuum type field equations, by passing to the spatio-temporal limit as $\Delta t \rightarrow 0, \Delta x \rightarrow 0$ make somewhat unrealistic assumptions in order to obtain tractable partial differential equations to enable qualitative estimates of the true population behavior. One must question the process of first homogenizing an inherently discrete population's characteristics in order to develop PDE-based continuum models and then re-discretizing them into nodal values. This process is not bijective, in other words, one does not recover the original discrete system (Fig. 2). Information is lost in this process. Also, because of the simplifying assumptions on interaction, births, age structuring, etc., that are typically made, in order to obtain tractable field equations, the resulting discrete equations are not as physically meaningful as the true discrete population interaction that it is based upon. Essentially, in dealing with small subpopulations, or populations with complex mixing and high heterogeneity, the assumptions behind regularization techniques leading to continuum models may be difficult to justify. This motivates developing agent based methods, which are based on direct accounting of interactions between individuals or population subgroups.

1.4 Agent-based models and objectives

Agent based models attempt to model the interaction between individuals directly, incorporating stochastic methods and have been used across many different disciplines, such as: biology, business, economics, social sciences, robotics and technology. The objective is to obtain deeper insight into the connection between micro-interaction and emergent macroscale behavior. The method first appeared in the 1940's, but was computationally infeasible until the 1990's. Although there are many forms of agent-based models, computationally, they are quite similar to particle interaction models in mathematical physics. The use of the term agent is often attributed to Holland and Miller's 1991 paper "Artificial Adaptive Agents in Economic Theory" [33]. The 1990's and early 2000's led to many approaches, often correlated to the concepts of swarms of agents and aggregate movement. For reviews of the literature, see Bonabeau [11]. The attractive features of the approaches are the ability to model individual interactions nonphenomenologically, and to allow the system to evolve autonomously. The utility of such approaches is that one can trivially modify the "rules of engagement", population sizes, reproduction rates, etc, and provide quantitative spatial and temporal information. Clearly, such a computational technique is easy to implement, and it is no extra effort to increase the number of population character parameters. We refer the reader to Zohdi [49,50,53–55], for reviews. Our objective in the current paper is to apply similar concepts to pandemic modeling, where the interaction is via infection transmission. Specifically, we develop a robust agent-based computational framework to investigate the emergent structure of Susceptible-Infected-Removed/Recovered (SIR)-type populations and more complex extensions on a global planetary scale, where each agent on the surface of the planet is initially uninfected. Infections are then seeded on the planet in localized regions. Contracting an infection depends on the characteristics of each agent—i.e. their susceptibility and contact with the seeded, infected, agents. Agent mobility on the planet is dictated by social policies, for example such as "shelter in place", "complete lockdown", etc. The global population is then allowed to evolve according to infected states of agents, over many time periods, leading to an SIR population. The work illustrates the construction of the computational framework and provides numerical examples.

Remarks: In the related field of astro-biology, there are concerns about Earth-based microbes potentially infecting other worlds, by being carried on spacecraft missions. The recent discovery of water on Mars (as well as the moons of Saturn and Jupiter) has heightened such concerns. It has been reported that, on Mars, liquid water can possibly form seasonally in locations where snow is present on soils, in the presence of saline, producing brine (Martinez and Renno [37]). Since terrestrial bacteria can grow brine, infection of the Marian biosphere could be possible. The reverse is also true, since returning spacecraft from Mars could bring back non-terrestrial organisms back to Earth. NASA and other space agencies have consistently reported that Bacillus spores, subjected to years vacuum in space, cosmic radiation and extreme temperatures, can survive, if they are shielded by the exterior of a spacecraft. We refer readers to Beaty [6], Fischer et al. [23], Martinez and Renno [37], Summons et al. [46], Michalski [39] and Debus [18] for details.

2 Direct agent-based interaction models

Following Zohdi [53], we now construct a model problem based on discrete rule-driven interaction between agents of subpopulations. One can consider an agent as an individual or a small group of individuals (a "meta-individual").

2.1 Agent-to-agent interaction and rules of engagement

Consider the following construction, for the "rules of engagement" for intermeshed infected and uninfected subpopulations, which are in close proximity to one another (Fig. 4):

• If two agents of the subpopulations, denoted (I = infected) and (U = uninfected), come within a certain distance,

$$||\boldsymbol{r}_{i}^{(I)} - \boldsymbol{r}_{j}^{(U)}|| \le d_{IU},$$
(2.1)

then two are said to engage in "contact".

• If the uninfected agent is susceptible, then the agent becomes infected.



Fig. 4 The infection zone and flow chart

- The "susceptibility" of the uninfected subpopulations is heterogeneous (preset at the beginning of the simulation).
- The incubation period for an infected agent to either recover or die is *T*_{*I*}.
- Once an agent of the population perishes or becomes immune, it cannot affect the rest of the population.

2.2 Algorithm

The algorithm is as follows:

- Step 1: Select:
- (a) The number of agents (N_a) in the populations.
- (b) The infection distance.
- (c) The susceptibility of the agents.
- (d) The total simulation time T.
- (e) The cycle time = Δt . The number of time-periods is $\frac{T}{\Delta t}$.



Fig. 5 Spherical coordinates used for agent placement. In this example, agent high-density concentration centers are at the (z-poles)

- Step 2: Generate the initial population locations on the globe.
- Step 3: For each population, loop over each agent in the infection radius. If so, according to the "rules of engagement" in the previous section, compute the interaction of the pair.
- Step 4: Compute the survivors and deaths of the existing agents for the time period.
- Step 5: Repeat steps 2–4 for the next time period until the overall simulation time is complete.

The relative ease at which one can generate such a population, and step it through several time periods, is rather obvious.

2.3 Computational acceleration

There are a variety of techniques to accelerate the computations. The primary computational expense is neighbor-toneighbor contact checks (an $O(N_a^2)$ operation). To mitigate this, one can construct so-called "interaction" or "Verlet" lists of neighboring agents that an agent may be in contact with at any given time. One can retain this Verlet list for a preset number of time steps and then update it when appropriate. The approach is relatively straightforward to implement and can speed up the computations dramatically (see Pöschel and Schwager [45] and Zohdi [51,52]). Alternative computational acceleration approaches can be achieved via sorting and binning methods, which proceed by partitioning the whole domain into bins. The agents are sorted by the bins in which they reside. The agent interaction proceeds, bin by bin, where the agents within a bin potentially only interact with agents within their bin or neighboring bins. Parallel processing is a further acceleration that can be employed whereby groups of agents or bins are sent to each processor and updated periodically, sharing the agent information between processors every few time-steps. In this work, we only implemented the Verlet list algorithm.

3 A model problem

As an example, consider a population with 20,000 agents (Fig. 6). We consider six cases of increasingly mobile subpopulations that are initially uniformly mixed across the globe, with agents being infected or uninfected and being susceptible or not susceptible. We employed the following parameters:

- Globe radius, R = 1.
- Total simulation time: T = 30.
- The infection distance: $d_{IU} = 0.01R$.
- The population initially infected = 1%.
- The population that is susceptible = 25%.

- The incubation period for an agent to either recover or die is $T_I = 5$.
- Update cycle time = $\Delta t = 0.1$.
- The location of each agent was achieved using a random spherical coordinate scheme (see Fig. 9), whereby θ (inclination angle) is a random number between $0 \le \theta \le \pi$ and ϕ (azimuth angle) is a random number between $0 \le \phi \le 2\pi$, yielding the following in Cartesian coordinates:

$$r_x = Rsin(\theta(t=0))cos(\phi(t=0)), \qquad (3.1)$$

and

$$r_{y} = Rsin(\theta(t=0))sin(\phi(t=0)), \qquad (3.2)$$

and

$$r_z = R\cos(\phi(t=0)). \tag{3.3}$$

• The mobility is given by (the location at an instant of time later)

$$r_x = Rsin(\theta(t + \Delta t))cos(\phi(t + \Delta t))$$
(3.4)

and

$$r_{y} = Rsin(\theta(t + \Delta t))sin(\phi(t + \Delta t))$$
(3.5)

and

$$r_z = Rcos(\phi(t + \Delta t)). \tag{3.6}$$

This produces dense population centers at the z-poles. The updated values for the angles are given by

Inclination mobility: $\theta(t + \Delta t) = \theta(t) + \Delta \theta$, (3.7)

where $\Delta \theta = A_{\theta} \times \beta$, β being a random number between $-1 \le \beta \le 1$ and A_{θ} is a mobility amplitude parameter and

Azimuthal mobility:
$$\phi(t + \Delta t) = \phi(t) + \Delta \phi$$
, (3.8)

where $\Delta \phi = A_{\phi} \times \gamma$, γ being a random number between $-1 \le \gamma \le 1$ and A_{ϕ} is a mobility amplitude parameter.

Figure 6 illustrates, starting from left to right and top to bottom, the progressive growth of SIR subpopulations. Shown are results at t = 0, t = 0.2T, t = 0.4T, t = 0.6T, t = 0.8T and t = T time, with mobility parameters: $A_{\phi} = 0.004$ and $A_{\theta} = 0.004$. Figure 7 shows the evolution of infected and dead/recovered with mobility parameters: $0.0 \le A_{\phi} \le$

Computational Mechanics



Fig. 6 Starting from left to right and top to bottom, the progressive growth of at SIR population. Shown are after t = 0, t = 0.2T, t = 0.4T, t = 0.6T, t = 0.8T and t = T time, with mobility parameters: $A_{\phi} = 0.004$ and $A_{\theta} = 0.004$



Fig. 7 Evolution of infected and dead/recovered with mobility parameters: A_{ϕ} , $A_{\theta} = 0.000, 0.001, 0.002, 0.003, 0.004$ and 0.005

0.005 and $0 \le A_{\theta} \le 0.005$. With increases in mobility, there is an immediate and direct impact on the spread of the infection, emanating primarily from the dense population centers at the z-poles. The utility of the presented computational approach is that one can trivially modify the "rules of engagement", population sizes, etc., *and provide quantitative spatial and temporal information*. Clearly, *such a computational technique is easy to implement, and it is no extra effort to increase the number of population character parameters. This is straightforward to implement*. Numerous additional features can be added easily. For example, one could add population growth in a variety of ways, such as, algorithmically:

- If an agent of a population survives beyond a certain number of time periods, it then produces offspring, and then perishes.
- The offspring are placed within an "offspring" radius, centered at the spatial location of the parent. The number of children possible that an individual can have, at maturity, is given by

$$offspring = integer(\phi \times M)$$
 (3.9)

where $0 \le \phi \le 1$ is a random number and where *M* is the maximum number of children possible. The function "integer" extracts the nearest integer from $(\phi \times M)$.

• After giving birth to the offspring once, the agent cannot have offspring again.

We note that, if desired, incorporation of "forbidden regions" i.e. "uninhabitable zones" within the domain is relatively easily to enforce by checking at each time step whether an individual has entered such an area (Fig. 8). If so, then the individual is moved back outside, and a new position is recalculated with a different trajectory. Another extension to the overall modeling is to provide more detail on the movement of the agents. For example, as described at the outset of this paper, we could have allowed for them (as well as the parents) to move according to more physical rules, based on pandemic events and stimuli. It is here where one can draw on swarm movement models discussed earlier in the paper. This is discussed further next.

4 Extensions

One direction in which to extend this research is in the swarmlike movement of agents in response to pandemic events. The origins of swarm modeling are in the description of biological groups (flocks of birds, schools of fish, crowds of human beings, etc.) responses to predators or prey (Breder 1952



Fig. 8 Creating a "forbidden" zone

[13]). Early approaches that rely on decentralized organization can be found in Beni [8], Brooks [14], Dudek et al. [20], Cao et al. [16], Liu and Passino [36] and Turpin et al [47]. Usual models incorporate a tradeoff between long-range interaction and short-range repulsion between individuals, dependent on the relative distance between individuals (see Gazi and Passino [24–26], Bender and Fenton [7] or Kennedy and Eberhart [34]). The most basic model is to treat each individual as a point mass (Zohdi [49]), which we adopt here, and to allow the system to evolve, based on Newtonian mechanics, using a combination of short-range and long-range interaction forces (Gazi and Passino [24], Bender and Fenton [7], Kennedy and Eberhart [34] and Zohdi [49,50,53-55]).¹ For some creatures, the "visual field" of individuals may play a significant role, while if the agents are robots or Unmanned Autonomous Vehicles (UAVs) the communication can be electronic. However, in some systems, agents interact with a specific set of other agents, regardless of whether they are far away (Feder [21]). This appears to be the case for Starlings (Sturnus vulgaris). In Ballerini et al. [4], the authors concluded, that such birds communicate with a certain number of birds surrounding it and that interactions are governed by topological distance and not metric distance. Interested readers are referred to Ballerini et al. [4]. We refer the reader to Zohdi [49,50,53-55] for reviews and provide an example of such a formulation next.

4.1 An example of a swarm formulation

In order to illustrate how swarm movement is modeled, following Zohdi [49,50,53–55], we treat the agents as point

¹ There are other modeling paradigms, for example mimicking ant colonies (Bonabeau et al. [9]) which exhibit foraging-type behavior and trail-laying-trail-following mechanisms for finding food sources (see Kennedy and Eberhart [34] and Bonabeau et al. [9], Dorigo et al. [19], Bonabeau et al. [9], Bonabeau and Meyer [10] and Fiorelli et al. [22]).



Fig. 9 Agent-obstacle-target model problem

masses, i.e. we ignore their dimensions. For each agent (N_s in total) the equations of motion are

$$m_i \dot{\boldsymbol{v}}_i = m_i \ddot{\boldsymbol{r}}_i = \boldsymbol{\Psi}_i^{tot} = \mathcal{F}(N_i^{at}, N_i^{ao}, N_i^{aa}), \qquad (4.1)$$

where the position of a point (agent) in space is given by the vector \mathbf{r}_i , the velocity is given by $\mathbf{v}_i = \dot{\mathbf{r}}_i$, the acceleration is given by $\mathbf{a}_i = \dot{\mathbf{v}}_i = \ddot{\mathbf{r}}_i$, and where Ψ_i^{tot} represents the total forces acting on an agent *i*, N_i^{at} represents the interaction between agent *i* and desired targets, N_i^{ao} represents the interaction between agent *i* and obstacles and N_i^{aa} represents the interaction between agent *i* and other agents. In the context of a pandemic, examples of targets could be hospitals, food distribution centers, etc., while obstacles could be physical barriers, such as buildings.

4.1.1 Agent-target interaction

Consider agent-target interaction

$$||\mathbf{r}_{i} - \mathbf{T}_{j}|| = \left((r_{i1} - T_{j1})^{2} + (r_{i2} - T_{j2})^{2} + (r_{i3} - T_{j3})^{2}\right)^{1/2}$$

$$\stackrel{\text{def}}{=} d_{ij}^{at}, \qquad (4.2)$$

where T_j is the position vector to target j and the direction to each target is

$$\boldsymbol{n}_{i \to j} = \frac{\boldsymbol{T}_j - \boldsymbol{r}_i}{||\boldsymbol{r}_i - \boldsymbol{T}_j||}.$$
(4.3)

For each agent (i), we compute a weighted direction to each target

$$\hat{\boldsymbol{n}}_{i \to j} = (w_{t1}e^{-a_1 d_{ij}^{at}} - w_{t2}e^{-a_2 d_{ij}^{at}})\boldsymbol{n}_{i \to j}, \qquad (4.4)$$

where the w_{ti} are weights reflecting the importance of the target, a_i are decay parameters, which is summed (and normalized later in the analysis) to give an overall direction to move towards

$$N_i^{at} = \sum_{j=1}^{N_t} \hat{\boldsymbol{n}}_{i \to j}.$$
(4.5)

4.1.2 Agent-obstacle interaction

Now consider agent-obstacle interaction

$$||\mathbf{r}_{i} - \mathbf{O}_{j}|| = \left((r_{i1} - O_{j1})^{2} + (r_{i2} - O_{j2})^{2} + (r_{i2} - O_{j2})^{2}\right)^{1/2}$$

$$\stackrel{\text{def}}{=} d_{ij}^{ao}, \qquad (4.6)$$

where O_j is the position vector to obstacle j and the direction to each obstacle is

$$\boldsymbol{n}_{i \to j} = \frac{\boldsymbol{O}_j - \boldsymbol{r}_i}{||\boldsymbol{r}_i - \boldsymbol{O}_j||}.$$
(4.7)

For each agent (i), we compute a weighted direction to each obstacle

$$\hat{\boldsymbol{n}}_{i \to j} = (w_{o1}e^{-b_1 d_{ij}^{ao}} - w_{o2}e^{-b_2 d_{ij}^{ao}})\boldsymbol{n}_{i \to j},$$
(4.8)

where the w_{oi} are weights reflecting the importance of the obstacle, b_i are decay parameters, which is summed (and normalized later in the analysis) to give an overall direction to move towards

$$N_i^{ao} = \sum_{j=1}^{N_o} \hat{\boldsymbol{n}}_{i \to j}.$$
(4.9)

4.1.3 Agent–agent interaction

Now consider agent(i)-agent(j) interaction

$$||\mathbf{r}_{i} - \mathbf{r}_{j}|| = \left((r_{i1} - r_{j1})^{2} + (r_{i2} - r_{j2})^{2} + (r_{i3} - r_{j3})^{2}\right)^{1/2}$$

$$\stackrel{\text{def}}{=} d_{ij}^{aa}, \qquad (4.10)$$

and the direction to each agent

$$\boldsymbol{n}_{i \to j} = \frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{||\boldsymbol{r}_i - \boldsymbol{r}_j||}.$$
(4.11)

For each agent (i), we compute a weighted direction to each agent

$$\hat{\boldsymbol{n}}_{i \to j} = (w_{a1}e^{-c_1 d_{ij}^{aa}} - w_{a2}e^{-c_2 d_{ij}^{aa}})\boldsymbol{n}_{i \to j}, \qquad (4.12)$$

where the w_{ai} are weights reflecting the importance of the agents, c_i are decay parameters, which is summed (and normalized later in the analysis) to give an overall direction to move towards

$$N_{i}^{aa} = \sum_{j=1}^{N_{a}} \hat{n}_{i \to j}.$$
(4.13)

4.1.4 Summation of interactions

We now aggregate the contributions by weighting their overall importance with weights for agent/target interaction, W_{at} , agent/obstacle interaction, W_{ao} and agent/agent interaction, W_{aa} :²

$$N_i^{tot} = W_{at} N_i^{at} + W_{ao} N_i^{ao} + W_{aa} N_i^{aa},$$
(4.14)

normalize the result

$$\boldsymbol{n}_{i}^{*} = \frac{N_{i}^{tot}}{||N_{i}^{tot}||}.$$
(4.15)

The forces are then constructed by multiplying the thrust force available by the propulsion system (foot, bicycle, car, etc.), F_i , by the overall normal direction

$$\Psi_i^{tot} = F_i \boldsymbol{n}_i^*. \tag{4.16}$$

We then integrate the equations of motion:

$$\boldsymbol{m}_i \boldsymbol{\dot{v}}_i = \boldsymbol{\Psi}_i^{tot}, \tag{4.17}$$

yielding

$$\boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i(t) + \frac{\Delta t}{m_i} \boldsymbol{\Psi}_i^{tot}(t)$$
(4.18)

and

$$\boldsymbol{r}_i(t + \Delta t) = \boldsymbol{r}_i(t) + \Delta t \boldsymbol{v}_i(t). \tag{4.19}$$

Note that if

$$||\boldsymbol{v}_i(t+\Delta t)|| > v_{max},\tag{4.20}$$

then we define $v_i^{old}(t + \Delta t) = v_i(t + \Delta t)$ and the velocity is rescaled

$$\boldsymbol{v}_i^{new}(t+\Delta t) = v_{max} \frac{\boldsymbol{v}_i^{old}(t+\Delta t)}{||\boldsymbol{v}_i^{old}(t+\Delta t)||},\tag{4.21}$$

with $\boldsymbol{v}_i(t + \Delta t) = \boldsymbol{v}_i^{new}(t + \Delta t)$.

5 An algorithm for movement in a region

Consider the following algorithm:

- 1. Initialize the locations of the targets: $T_i = (T_x, T_y, T_z)_i$, i = 1, 2,... N_T = targets.
- 2. Initialize the locations of the obstacles: $O_i = (O_x, O_y, O_z)_i$, $i = 1, 2, ..., N_O$ = obstacles.
- 3. Initialize the locations of the agents: $\mathbf{r}_i = (r_x, r_y, r_z)_i$, i = 1, 2,... N_a = agents.
- 4. For each agent (i), determine the distance and directed normal to each target, obstacle and other agents.
- 5. For each agent (i), determine interaction functions N_i^{at} , N_i^{ao} , N_i^{aa} and n_i^* .
- 6. For each agent (i), determine force acting upon it, $\Psi_i^{tot} = F_i n_i^*$.
- 7. For each agent (i), integrate the equations of motion (checking constraints) to produce $v_i(t + \Delta t)$ and $r_i(t + \Delta t)$.
- 8. Determine if any targets have been reached by checking the distance between agents and targets

$$||\boldsymbol{r}_i - \boldsymbol{T}_j|| \le Tolerance. \tag{5.1}$$

For any T_j , if any agent has satisfied this criteria, then immobilize *i* (fix r_i).

9. The entire process is then repeated for the next time step.

5.1 Preliminary numerical example

As a preliminary example, consider the following parameters:

- Mass = 75 kg,
- 100 agents,
- 1 target,
- 10 obstacles,
- T = 2000 s,
- $\Delta t = 0.001 \text{ s},$
- Initial agent velocity, $v_i(t = 0) = 0$ m/s,
- Initial agent domain (10, 10, 0) m,
- Thrust force available by the system, $F_i = 10^7$ Nt,
- Domain of (500, 500, 0) m,
- Maximum velocity agent $v_{max} = 2$ m/s.

The vector of system parameter inputs is

$$\begin{split} \mathbf{\Lambda}^{i} &\stackrel{\text{def}}{=} \{\Lambda_{1}, \Lambda_{2} ... \Lambda_{N}\} \\ &= \{W_{ma}, W_{ao}, W_{aa}, w_{t1}, w_{t2}, w_{o1}, w_{o2}, w_{a1}, \\ & w_{a2}, a_{1}, a_{2}, b_{1}, b_{2}, c_{1}, c_{2}\} \end{split}$$
(5.2)

² The parameters in the model will be optimized shortly.

was given by a test parameter vector

$$\mathbf{\Lambda}^{i} \stackrel{\text{def}}{=} \{10.00, 5.00, 7.50, 0.25, 0.50, 0.60, 0.80, 0.30, 0.85, 0.15, 0.50, 1.00, 0.75, 0.90, 0.60\},$$
(5.3)

selected within the following intervals:

- Overall weights: $0 \le W_{at}, W_{ao}, W_{aa} \le 10$,
- Target weights: $0 \le w_{t1}, w_{t2} \le 1$,
- Obstacle weights: $0 \le w_{o1}, w_{o2} \le 1$,
- Agent weights: $0 \le w_{a1}, w_{a2} \le 1$ and
- Decay coefficients: $0 \le a_1, a_2 \le 1, 0 \le b_1, b_2 \le 1, 0 \le c_1, c_2 \le 1$.

Figure 10 illustrates the results of this parameter choice.

5.2 Parameter estimation via machine-learning

There are many parameters in the system, warranting the use a Machine Learning Algorithm. Here we follow Zohdi [50,54,55] in order to optimize behavior by minimizing a cost function. For example, let us consider minimizing the following cost function

$$\Pi(\mathbf{\Lambda}) = \frac{(N_{tot} - N_a)}{N_{tot}}$$
(5.4)

where N_a represents the number agents that reached the target within a specific time and N_{tot} represents the total number of agents in the system. In other words, the system is being driven to the parameters generating the best case scenario (all agents reaching the target). The design vector of system parameters is:

$$\begin{aligned}
\mathbf{\Lambda} &= \{\Lambda_1, \Lambda_2 ... \Lambda_N\} \\
&= \{W_{mt}, W_{mo}, W_{mm}, w_{t1}, w_{t2}, w_{o1}, w_{o2}, w_{m1}, \\
& w_{m2}, a_1, a_2, b_1, b_2, c_1, c_2\}.
\end{aligned}$$
(5.5)

Cost functions associated with optimization of complex behavior are oftentimes nonconvex in design parameter space and often nonsmooth, as is the case for the system of interest. 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 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 (Holland [32]), as well as Goldberg [27], Davis [17], Onwubiko [43] and Goldberg and Deb [28]. The GA approach is extremely well-suited for nonconvex, nonsmooth, multicomponent, multistage systems, and involves the following essential concepts:

- 1. Population generation: Generate system population: $\mathbf{\Lambda}^{i} \stackrel{\text{def}}{=} \{\Lambda_{1}^{i}, \Lambda_{2}^{i}, \Lambda_{3}^{i}, \Lambda_{4}^{i}, ..., \Lambda_{N}^{i}\} = \{interaction \ parameters, ..., etc.\}^{i}$
- 2. *Performance evaluation*: Compute fitness/performance of each genetic string: $\Pi(\Lambda^i)$ and rank them (i = 1, ..., N)
- 3. *Mating process*: Mate pairs/produce offspring: $\lambda^i \stackrel{\text{def}}{=} \Phi^{(I)} \Lambda^i + (1 \Phi^{(I)}) \Lambda^{i+1}$ where $0 \le \Phi \le 1$ (Fig. 11)
- 4. *Gene elimination*: Eliminate poorly performing genetic strings, keep top parents and generated offspring
- 5. *Population regeneration*: Repeat the process with the new gene pool and new *random* genetic strings
- 6. Solution post-processing: Employ gradient-based methods afterwards in the local "valleys"-if smooth enough

5.2.1 Algorithmic specifics

Following Zohdi [50,54,55], the algorithm is as follows:

Step 1: Randomly generate a population of S starting genetic strings, Λⁱ, (i = 1, 2, 3, ..., S):

$$\begin{split} \mathbf{\Lambda}^{i} &\stackrel{\text{def}}{=} \{\Lambda_{1}, \Lambda_{2} \dots \Lambda_{N}\} \\ &= \{W_{mt}, W_{mo}, W_{mm}, w_{t1}, w_{t2}, w_{o1}, w_{o2}, w_{m1}, \\ & w_{m2}, a_{1}, a_{2}, b_{1}, b_{2}, c_{1}, c_{2}\}. \end{split}$$

- Step 2: Compute fitness of each string Π(Λⁱ), (i = 1, ..., S)
- *Step 3*: Rank genetic strings: Λ^i , (i = 1, ..., S)
- *Step 4*: Mate nearest pairs and produce two offspring, (i = 1, ..., S)

$$\boldsymbol{\lambda}^{i} \stackrel{\text{def}}{=} \boldsymbol{\Phi}^{(I)} \boldsymbol{\Lambda}^{i} + (1 - \boldsymbol{\Phi}^{(I)}) \boldsymbol{\Lambda}^{i+1},$$
$$\boldsymbol{\lambda}^{i+1} \stackrel{\text{def}}{=} \boldsymbol{\Phi}^{(II)} \boldsymbol{\Lambda}^{i} + (1 - \boldsymbol{\Phi}^{(II)}) \boldsymbol{\Lambda}^{i+1}$$

- *Step 5*: Eliminate the bottom *M* < *S* strings and keep top *K* < *S* 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
- *Note*: $\Phi^{(I)}$ and $\Phi^{(II)}$ are random numbers, such that $0 \le \Phi^{(I)} \le 1, 0 \le \Phi^{(II)} \le 1$, which are different for each component of each genetic string
- Option: Rescale and restart search around best performing parameter set every few generations



Fig. 10 Starting from left to right and top to bottom, the progressive movement of a group of agents (blue) avoiding obstacles (green) to get to the target (red)

• *Remark*: 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.

As another example, one can use such algorithms to search for parameter sets that yield subpopulations (such as infected (I) and recovered (R)) having similar stable sizes after many time-periods. Mathematically speaking, this can be expressed by writing min_{Λ} $\Pi(\Lambda)$, where

$$\Pi(\mathbf{\Lambda}) = \frac{|I - R|}{I + R},\tag{5.6}$$

and where $\Lambda \stackrel{\text{def}}{=} \{d_{IU}, A_{\phi}, A_{\theta}, T_I, etc.\}$, after a set number of time periods for a given Λ . Clearly, there are numerous possibilities. Typically, for populations with a finite number of agents, there will be slight variations in the performance for different random starting configurations. In order to stabilize the objective function's value with respect to the randomness of the starting configuration, for a given parameter selection (Λ), a regularization procedure is applied within the genetic algorithm, whereby the performances of a series of different



Fig. 11 The basic action of a machine learning-based genetic algorithm (Zohdi [55])

random starting configurations are averaged until the (ensemble) average converges, i.e. until the following condition is met:

$$\left| \frac{1}{Z+1} \sum_{i=1}^{Z+1} \Pi^{(i)}(\mathbf{\Lambda}^{I}) - \frac{1}{Z} \sum_{i=1}^{Z} \Pi^{(i)}(\mathbf{\Lambda}^{I}) \right| \\ \leq TOL\left(\frac{1}{Z+1} \sum_{i=1}^{Z+1} \Pi^{(i)}(\mathbf{\Lambda}^{I}) \right),$$
(5.7)

where index *i* indicates a different starting random configuration (i = 1, 2, ..., Z) that has been generated and Z indicates the total number of configurations tested. In order to implement this in the genetic algorithm, in step 2, one simply replaces *compute* with *ensemble compute*, which requires a further inner loop to test the performance of multiple starting configurations. Development of such stochastic Machine Learning Algorithms for complex, multiscale, systems is currently under investigation by the author.

References

- Altizer S, Nunn C (2006) Infectious diseases in primates: behavior, ecology and evolution. Oxford series in ecology and evolution. Oxford University Press, Oxford ISBN: 0-19-856585-2
- Anderson RM (ed) (1982) Population dynamics of infectious diseases: theory and applications. Chapman and Hall, London ISBN: 0-412-21610-8
- Bailey Norman TJ (1975) The mathematical theory of infectious diseases and its applications, 2nd edn. Griffin, London ISBN: 0-85264-231-8
- Ballerini M, Cabibbo N, Candelier R, Cavagna A, Cisbani E, Giardina I, Lecomte V, Orlandi A, Parisi G, Procaccini A, Viale M, Zdravkovic V (2008) Interaction ruling animal collective behavior depends on topological rather than metric distance: evidence from a field study. PNAS 105(4):1232–1237
- Bartlett MS (1957) Measles periodicity and community size. J R Stat Soc Ser A 120(1):48–70. https://doi.org/10.2307/2342553
- Beaty D, Buxbaum K, Meyer M, Barlow N, Boynton W, Clark B, Deming J, Doran PT (2006) Findings of the mars special regions

science analysis group. Astrobiology 6(5):677–732. https://doi. org/10.1089/ast.2006.6.677

- Bender J, Fenton R (1970) On the flow capacity of automated highways. Transp Sci 4:52–63
- 8. Beni G (1988) The concept of cellular robotic system. In: IEEE international symposium on intelligent control, pp 57–62
- Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press, New York
- Bonabeau E, Meyer C (2001) Swarm intelligence: a whole new way to think about business. Harv Bus Rev 79(5):106–114
- Bonabeau E (2002) Agent-based modeling: methods and techniques for simulating human systems. Proc Natl Acad Sci USA 99:7280–7287. https://doi.org/10.1073/pnas.082080899
- Brauer F, Castillo-Chavez C (2001) Mathematical models in population biology and epidemiology. Springer, New York ISBN: 0-387-98902-1
- Breder CM (1954) Equations descriptive of fish schools and other animal aggregations. Ecology 35(3):361–370
- Brooks RA (1991) Intelligence without reason. In: Proceedings of the international joint conference on artificial intelligence (IJCAI-91), pp 569–595
- Capasso V (1993) Mathematical structure of epidemic systems. Springer, Berlin ISBN: 3-540-56526-4
- Cao YU, Fukunaga AS, Kahng A (1997) Cooperative mobile robotics: antecedents and directions. Auton Robots 4(1):7–27
- 17. Davis L (1991) Handbook of genetic algorithms. Thompson Computer Press
- Debus A (2005) Estimation and assessment of Mars contamination. Adv Space Res 35(9):164853. https://doi.org/10.1016/j.asr.2005. 04.084
- Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 26(1):29–41
- Dudek G, Jenkin M, Milios E, Wilkes D (1996) A taxonomy for multi-agent robotics. Auton Robots 3:375–397
- 21. Feder T (2007) Statistical physics is for the birds. Phys Today 60:28–29
- Fiorelli E, Leonard NE, Bhatta P, Paley D, Bachmayer R, Fratantoni DM (2004) Multi-auv control and adaptive sampling in monterey bay. In: Autonomous underwater vehicles, 2004 IEEE/OES, pp 134–147
- Fischer E, Martinez GM, Elliott HM, Renno NO (2014) Experimental evidence for the formation of liquid saline water on Mars. Geophys Res Lett. https://doi.org/10.1002/2014GL060302
- Gazi V, Passino KM (2002) Stability analysis of swarms. In: Proceedings of the American control conference. Anchorage, AK May 8-10
- Gazi V, Passino KM (2003) Stability analysis of swarms. IEEE Trans Autom Control 48(4):692–697
- Gazi V, Passino KM (2004) A class of attraction/repulsion functions for stable swarm aggregation. Int J Control 77(18):1567–1579
- 27. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Boston
- Goldberg DE, Deb K (2000) Special issue on genetic algorithms. Comput Methods Appl Mech Eng 186(2–4):121–124
- Harko T, Lobo FSN, Mak MK (2014) Exact analytical solutions of the susceptible-infected-recovered (SIR) epidemic model and of the SIR model with equal death and birth rates. Appl Math Comput 236:184–194. https://doi.org/10.1016/j.amc.2014.03.030
- Hethcote H (2000) The mathematics of infectious diseases. SIAM Rev 42(4):599–653. https://doi.org/10.1137/s0036144500371907
- Hethcote Herbert W (1989) Three basic epidemiological models. In: Levin Simon A, Hallam Thomas G, Gross Louis J (eds) Applied mathematical ecology: biomathematics, vol 18. Springer,

Berlin, pp 119–144. https://doi.org/10.1007/978-3-642-61317-3_5 ISBN: 3-540-19465-7

- 32. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
- Holland JH, Miller JH (1991) Artificial adaptive agents in economic theory (PDF). Am Econ Rev 81(2):365–371 (Archived from the original (PDF) on October 27, 2005)
- 34. Kennedy J, Eberhart R (2001) Swarm intelligence. Morgan Kaufmann Publishers, Burlington
- Kermack WO, McKendrick AG (1927) A contribution to the mathematical theory of epidemics. Proc R Soc A 115(772):700–721. https://doi.org/10.1098/rspa.1927.0118
- 36. Liu Y, Passino KM (2000) Swarm intelligence: literature overview. Technical report, Ohio State University
- Martinez GM, Renno NO (2013) Water and brines on Mars: current evidence and implications for MSL. Space Sci Rev 175:29–51
- May Robert M, Anderson Roy M (1991) Infectious diseases of humans: dynamics and control. Oxford University Press, Oxford ISBN: 0-19-854040-X
- Michalski JR, Cuadros J, Niles PB, Parnell J, Rogers DA, Wright SP (2013) Groundwater activity on Mars and implications for a deep biosphere. Nat Geosci 6(2):133–138. https://doi.org/10.1038/ ngeo1706
- Miller JC (2012) A note on the derivation of epidemic final sizes. Bull Math Biol 74(9):2125–2141. https://doi.org/10.1007/s11538-012-9749-6
- Miller JC (2017) Mathematical models of SIR disease spread with combined non-sexual and sexual transmission routes. Infect Dis Model 2(1):33–55. https://doi.org/10.1016/j.idm.2016.12.003
- 42. Murray JD (2004) Mathematical biology, 3rd edn. Springer, Berlin
- 43. Onwubiko C (2000) Introduction to engineering design optimization. Prentice Hall, Upper Saddle River
- 44. Osemwinyen A, Diakhaby A (2015) Mathematical modelling of the transmission dynamics of ebola virus. Appl Comput Math 4:313–320 (The first and second differential equations are transformed and brought to the same form as for the SIR model above)

- 45. Pöschel T, Schwager T (2004) Computational granular dynamics. Springer, Berlin
- 46. Summons RE, Amend JP, Bish D, Buick R, Cody GD, Des Marais DJ, Dromart G, Eigenbrode JL et al (2011) Preservation of Martian organic and environmental records: final report of the Mars Biosignature Working Group. Astrobiology 11(2):157–181. https://doi. org/10.1089/ast.2010.0506
- Turpin M, Michael N, Kumar V (2014) Concurrent assignment and planning of trajectories for multiple robots. Int J Robot Res. https:// doi.org/10.1177/0278364913515307
- Vynnycky E, White RG (eds) (2010) An introduction to infectious disease modelling. Oxford University Press, Oxford ISBN: 978-0-19-856576-5
- Zohdi TI (2003) Computational design of swarms. Int J Numer Methods Eng 57:2205–2219
- Zohdi TI (2009) Mechanistic modeling of swarms. Comput Methods Appl Mech Eng 198(21–26):2039–2051
- Zohdi TI (2014) A direct particle-based computational framework for electrically-enhanced thermo-mechanical sintering of powdered materials. Math Mech Solids 19(1):93–113
- Zohdi TI (2014) Additive particle deposition and selective laser processing: a computational manufacturing framework. Comput Mech 54:171–191
- Zohdi TI (2016) An agent-based computational framework for simulation of competing hostile planet-wide populations. Comput Methods Appl Mech Eng. https://doi.org/10.1016/j.cma.2016.02. 035
- Zohdi TI (2018) Multiple UAVs for mapping: a review of basic modeling, simulation and applications. Ann Rev Environ Resour. https://doi.org/10.1146/annurev-environ-102017-025912
- Zohdi TI (2019) The game of drones: rapid agent-based machinelearning models for multi-UAV path planning. Comput Mech. https://doi.org/10.1007/s00466-019-01761-9

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.