

# MODELING WARRING FUNGI ON A PLANT.

STEVEN GLASFORD

**ABSTRACT.** In the wild, it is widespread for fungi and other pathogens to enter plant hosts and typically result in competition from other pathogens for a single host. Also typical is when a fungus mutates, whereby natural selection or by the act of introduction into a different environment. The mutation leads to a case where the fungus that has not mutated (the resident) might have to compete with its mutated cousin. This paper investigates a zero-sum game between a resident and mutant strain of fungi to see if one fungus wins over the other. We also explore the case in which the two strains enter an equilibrium point where both can exist. Furthermore, this paper simulates this situation on a single plant for a single season and does not discuss the potential for evolution between seasons or the dynamics of multiple plants of the same species.

## 1. INTRODUCTION

Pathogens are what make people, pets, and every other living organism sick. They are a broad group and can do everything from making a person sick with COVID-19 to blight rotting potatoes in the field starving people. Plant pathogens are everywhere. Going outside to the park, it is not uncommon for a person to see brown spots on the leaves of trees caused by fungus or to see lichens growing on oak trees' bark.

In this paper, a zero-sum feedback game [A.1] of two competing pathogens on a single plant, battling for dominance on the plant. To do this, we use a system of nonlinear ordinary differential equations. We use computers and numerical analysis to solve and describe the dynamics of a single growing season of two invading fungal pathogens on a single plant.

The first fungus can be understood as the resident, while the second is the mutant. The mutant fungus is of the same species but is a different individual. The function that describes the fungi invasion is the difference between the two marginal fitness criteria [A.10], and represents the cost in defining the differential game's value [A.9]. A general dynamic game approach of [15] is explored and discussed. The Cauchy problem [A.6] for the first-order Hamilton–Jacobi partial differential equation (HJ PDE) created by this situation is solved using the ROC-HJ software [4].

Analysis of the numerical simulations obtained by this software package, and a graphical description is further investigated later in this document.

## 2. STATEMENT OF THE MODEL

Before we can adequately analyze the numerical solutions, we must first define the model. We consider the dynamics of two biotrophic [A.7] fungal cohorts [A.6] on one plant in a single season. Let us refer to them as cohorts 1 and 2. For cohort  $i$ , let  $n_i$  the lesion density, which is the number of mycelia [A.5] per unit area of the host. For simplicity, assume  $n_1$  and

---

*Date:* Fall 2020.

*Advisor:* Dr. Ivan Yegorov.

$n_2$  to be constant during the entire infection period within the season, which means that only the resident and the mutant exist on the plant. There are no new invaders that do not penetrate the host during the analyzed period. In the wild, this is a substantial restriction for infections. However, it still allows us to draw significant conclusions with relevant biological interpretations while adhering to standard experimental infection protocols.

We denote the average size of a mycelium in cohort  $i$  by  $M_i$ . Let  $S_i$  be the average quantity of spores produced by a mycelium in cohort  $i$ . The duration of the infection  $t$  within the season is a time variable. Note that the mycelial sizes  $M_1$  and  $M_2$  are measured in terms of equivalent amounts of infecting spores. For instance, if a mycelium appears from one spore at the beginning of the infection period, the initial mycelial size can be set as one or only as one spore.

To determine the nutrient flux [A.8] of cohort  $i$ , we use the function,  $f_i = f_i(M_1, M_2)$ , where the flux allocation is between mycelia growth and producing spores. which is allocated between two different pathogen activities such as within-host multiplication (mycelial growth) and producing asexual spores. For brevity, we do not explicitly show the dependence on the constant parameters  $n_1, n_2$  when writing  $f_i = f_i(M_1, M_2)$ ,  $i = 1, 2$ . Let  $u_i = u_i(t)$  be a related resource allocation (control) function, taking values between zero and one. When  $u_i(t) = 0$ , the particular fungus only produces mycelia, while when  $u_i(t) = 1$ , the fungus produces spores, and when  $0 < u_i(t) < 1$ , the fungus is doing both to some degree.

Let  $g = g(M)$  be the rate of mycelial decay for both cohorts; we assume that the mutation does not. We assume a constant yield of spores,  $\delta > 0$ , as compared to mycelial growth.

The period of the infections within the season is a fixed interval  $[0, T]$ .

The following dynamic model can thus be formulated [15]:

$$(1) \quad \begin{cases} \frac{dM_1(t)}{dt} = (1 - u_1(t)) f_1(M_1(t), M_2(t)) - g(M_1(t)), \\ \frac{dM_2(t)}{dt} = (1 - u_2(t)) f_2(M_1(t), M_2(t)) - g(M_2(t)), \\ \frac{dS_1(t)}{dt} = \delta u_1(t) f_1(M_1(t), M_2(t)), \\ \frac{dS_2(t)}{dt} = \delta u_2(t) f_2(M_1(t), M_2(t)), \\ M_1(0) = M_1^0, \quad M_2(0) = M_2^0, \quad S_1(0) = S_1^0, \quad S_2(0) = S_2^0, \\ 0 \leq u_1(t) \leq 1, \quad 0 \leq u_2(t) \leq 1, \quad t \in [0, T]. \end{cases}$$

Since the right-hand sides of the third and fourth equations in (1) do not contain  $S_1$  and  $S_2$ , one does not need to treat  $S_1$  and  $S_2$  explicitly, which makes  $M_1$  and  $M_2$  considered only as state variables [A.11].

The reproductive success (the amount of growth on the plant) of cohort  $i$  is determined by:

$$(2) \quad \int_0^T \frac{dS_i(t)}{dt} e^{-\mu t} dt = \delta \int_0^T u_i(t) f_i(M_1(t), M_2(t)) e^{-\mu t} dt,$$

where  $e^{-\mu t}$  describes the exponential extinction of the infections as the plant develops immunity, and  $\mu$  is a positive constant. Since  $\delta$  is a positive constant, we can divide (2) by  $\delta$  and arrive at the following, where  $J$  is a Jacobi function,

$$J_i(u_1(\cdot), u_2(\cdot)) = \int_0^T u_i(t) f_i(M_1(t), M_2(t)) e^{-\mu t} dt.$$

In the next equations, we look at the zero-sum two-player differential game, in line with [3, 15]. We look at the competition between the two cohorts in a single season by investigating their reproductive outputs and seeking the saddle control strategies [A.12]. In other words, we try and find the point/s in which the two pathogens enter a stalemate and what actions the fungi need to do in order to enter this situation.

$$(3) \quad \begin{cases} \frac{dM_1(t)}{dt} = (1 - u_1(t)) f_1(M_1(t), M_2(t)) - g(M_1(t)), \\ \frac{dM_2(t)}{dt} = (1 - u_2(t)) f_2(M_1(t), M_2(t)) - g(M_2(t)), \\ M_1(0) = M_1^0, \quad M_2(0) = M_2^0, \\ 0 \leq u_1(t) \leq 1, \quad 0 \leq u_2(t) \leq 1, \quad t \in [0, T], \end{cases}$$

$$(4) \quad J(u_1(\cdot), u_2(\cdot)) \longrightarrow \inf_{u_1(\cdot)} \sup_{u_2(\cdot)} \text{ or } \sup_{u_2(\cdot)} \inf_{u_1(\cdot)},$$

$$(5) \quad \begin{aligned} J(u_1(\cdot), u_2(\cdot)) &= J_2(u_1(\cdot), u_2(\cdot)) - J_1(u_1(\cdot), u_2(\cdot)) \\ &= \int_0^T (u_2(t) f_2(M_1(t), M_2(t)) - u_1(t) f_1(M_1(t), M_2(t))) e^{-\mu t} dt. \end{aligned}$$

Where inf is the infimum [B.1] and sup is the supremum [B.2].

According to [15], we represent the nutrient fluxes [A.8] and decay rates as

$$(6) \quad f_i(M_1, M_2) = \nu(n_1 M_1 + n_2 M_2) \cdot \rho(M_i) \quad \forall M_1 \geq 0 \quad \forall M_2 \geq 0, \quad i = 1, 2,$$

$$(7) \quad \rho(M) = \alpha \frac{M}{M + k} \quad \forall M \geq 0,$$

$$(8) \quad \nu(n_1 M_1 + n_2 M_2) = \frac{1}{1 + \beta(n_1 M_1 + n_2 M_2)} \quad \forall M_1 \geq 0 \quad \forall M_2 \geq 0,$$

$$(9) \quad g(M) = \gamma M \quad \forall M \geq 0.$$

$\rho$  is a positive function that specifies the flow of resources obtained by a single mycelium, and  $\nu$  describes the negative consequences as the pathogens compete for resources from the single plant [15].

Furthermore, the following parameter values are also useful for the numerical simulations and arise from the analysis of [16]:

$$(10) \quad \begin{aligned} \alpha &= 0.2 \cdot 10^4 \text{ spores/day}, & k &= (1/6) \cdot 10^4 \text{ spores}, \\ \beta &= 10^{-5} \text{ cm}^2/\text{spores}^2, & \gamma &= 0.06 \text{ 1/day}, & \mu &= 0.03 \text{ 1/day}, \\ n_1 &= 9 \text{ spores/cm}^2, & n_2 &= 1 \text{ spore/cm}^2, \\ T &= 60 \text{ days}. \end{aligned}$$

As was shown in [15], the bounded domain

$$G = \{(M_1, M_2) \in \mathbb{R}^2 : 0 < M_1 < \bar{M}_1, 0 < M_2 < \bar{M}_2\}$$

with

$$\bar{M}_1 = \bar{M}_2 > \alpha/\gamma - k$$

is an invariant set in the state space (if a state trajectory starts in  $G$ , it cannot leave  $G$ ). The parameters  $\bar{M}_1, \bar{M}_2$  can be understood as carrying capacity [A.13] estimates.

### 3. UNINVADABLE AND EVOLUTIONARILY STABLE STRATEGIES

In war, there are times in which neither party wins. Where there is a boundary or other concessions exist, the resident is uninvadable under this circumstance. This section describes the strategies and functions used to model a stable and uninvadable resident pathogen. In other words, we see where the two pathogens enter a stalemate, and we look deeper at the game-theoretic statements from (3)–(5).

Based on the terminology of Adaptive Dynamics [8], let us interpret cohort 1 as a resident and cohort 2 as a mutant. Denote the corresponding classes of considered strategies as  $\mathcal{U}_1$  and  $\mathcal{U}_2$ . For a pair of methods  $(u_1, u_2) \in \mathcal{U}_1 \times \mathcal{U}_2$ , the resident is not invaded by the mutant if and only if

$$J(u_1, u_2) = J_2(u_1, u_2) - J_1(u_1, u_2) \leq 0.$$

A strategy  $\hat{u}_1 \in \mathcal{U}_1$  is hence uninvadable, and the winner of the war is the resident if and only if

$$J(\hat{u}_1, u_2) \leq 0 \quad \forall u_2 \in \mathcal{U}_2,$$

which is equivalent to

$$\sup_{u_2 \in \mathcal{U}_2} J(\hat{u}_1, u_2) \leq 0$$

where  $\sup$  is the supremum [B.2].

Such a  $\hat{u}_1$  exists if

$$(11) \quad \inf_{u_1 \in \mathcal{U}_1} \sup_{u_2 \in \mathcal{U}_2} J(u_1, u_2) = \min_{u_1 \in \mathcal{U}_1} \sup_{u_2 \in \mathcal{U}_2} J(u_1, u_2) \leq 0$$

or

$$\inf_{u_1 \in \mathcal{U}_1} \sup_{u_2 \in \mathcal{U}_2} J(u_1, u_2) < 0$$

(The latter inequality arises from when the infimum [B.1]  $u_1$  is not satisfied). Equation (11) motivates our game-theoretic statement (3)–(5), where the first player tries to maximize its resistance to an invasion by the second one, and vice versa. In other words, the resident fungus tries to grow and fight the opposing mutant fungus, and the mutant fungus tries to do the same.

If  $M_1^0 = M_2^0$  and  $\mathcal{U}_1 = \mathcal{U}_2 = \mathcal{U}$ , then  $J(u, u) = 0$  for all  $u \in \mathcal{U}$ , and (11) simplifies to

$$\min_{u_1 \in \mathcal{U}} \sup_{u_2 \in \mathcal{U}} J(u_1, u_2) = 0$$

This equation is the boundary where neither the resident and the mutant win, and there is a stalemate. In this case, a strategy

$$\hat{u}_1 \in \operatorname{Arg} \min_{u_1 \in \mathcal{U}} \left( \sup_{u_2 \in \mathcal{U}} J(u_1, u_2) \right)$$

where  $\operatorname{Arg} \min$  is the argument of the minimum [B.3]. The strategy is called evolutionary stable, and there is a boundary where neither pathogen wins if the related maximum for  $u_2$  is unique:

$$\operatorname{Arg} \max_{u_2 \in \mathcal{U}} J(\hat{u}_1, u_2) = \{\hat{u}_1\}.$$

where  $\operatorname{Arg} \max$  is the argument of the maximum [B.3].

This approach to defining stable evolutionary strategies was initially proposed in [3] and further developed in [15].

#### 4. HAMILTON–JACOBI–ISAACS EQUATION

To determine how evolutionary strategies change over time, we must develop a system of differential functions. This circumstance can be solved using a Hamilton–Jacobi–Isaacs equation. We exclude the reasoning behind using this for brevity, [15] contains more information and further analysis.

We introduce the following function known as Hamiltonian:

$$\begin{aligned} H(t, M_1, M_2, u_1, u_2, p_1, p_2) &= p_1 ((1 - u_1) f_1(M_1, M_2) - g(M_1)) \\ &\quad + p_2 ((1 - u_2) f_2(M_1, M_2) - g(M_2)) \\ &\quad + e^{-\mu t} (u_2 f_2(M_1, M_2) - u_1 f_1(M_1, M_2)) \end{aligned}$$

$$\forall t \in [0, T] \quad \forall (M_1, M_2) \in G \quad \forall (u_1, u_2) \in [0, 1]^2 \quad \forall (p_1, p_2) \in \mathbb{R}^2$$

(this is the sum of the dot product of  $p = (p_1, p_2)$  with  $(dM_1/dt, dM_2/dt)$  and the integrand in  $J(u_1(\cdot), u_2(\cdot))$ , see (3) and (5)). The Hamiltonian satisfies the saddle point condition [B.4] concerning the control variables  $u_1$  and  $u_2$

$$\begin{aligned} \min_{u_1 \in [0, 1]} \max_{u_2 \in [0, 1]} H(t, M_1, M_2, u_1, u_2, p_1, p_2) \\ = \max_{u_2 \in [0, 1]} \min_{u_1 \in [0, 1]} H(t, M_1, M_2, u_1, u_2, p_1, p_2) = \mathcal{H}(t, M_1, M_2, p_1, p_2) \end{aligned}$$

$$\forall t \in [0, T] \quad \forall (M_1, M_2) \in G \quad \forall (p_1, p_2) \in \mathbb{R}^2$$

(here minimax and maximin [B.5] give the same result). Due to the theoretical results of [9, §XI.6], the value function  $V: [0, T] \times G \rightarrow \mathbb{R}$  in the formulated feedback differential game is a unique solution of the following Cauchy problem [B.6] for the Hamilton–Jacobi–Isaacs partial differential equation (HJI PDE):

$$(12) \quad \begin{cases} \frac{\partial V(t, M_1, M_2)}{\partial t} + \mathcal{H}\left(t, M_1, M_2, \frac{\partial V(t, M_1, M_2)}{\partial M_1}, \frac{\partial V(t, M_1, M_2)}{\partial M_2}\right) = 0, \\ V(T, M_1, M_2) = 0, \\ t \in [0, T], \quad (M_1, M_2) \in G. \end{cases}$$

In general, the solution may be nonsmooth [B.7] and understood in a generalized (viscosity [B.8] or minimax) sense [9, 14]. These conditions make it difficult, if not impossible, to find an exact solution.

The general properties of value functions of zero-sum feedback differential games [6] limit how fast the value function  $V$  changes. These limitations make it Lipschitz continuous [B.9] in  $[0, T] \times G$ . Rademacher's theorem [B.10] is differentiable almost everywhere in  $(0, T) \times G$ , except for possibly a subset of Lebesgue measure zero.

For points  $(t, M_1, M_2)$  at which the value function is differentiable, one can obtain the saddle feedback control strategies (where the fungi enter a stalemate) from the relations

$$\begin{aligned} u_1(t, M_1, M_2) &\in \operatorname{Arg} \min_{w_1 \in [0,1]} \left\{ \max_{w_2 \in [0,1]} H \left( t, M_1, M_2, w_1, w_2, \frac{\partial V(t, M_1, M_2)}{\partial M_1}, \frac{\partial V(t, M_1, M_2)}{\partial M_2} \right) \right\}, \\ u_2(t, M_1, M_2) &\in \operatorname{Arg} \max_{w_2 \in [0,1]} \left\{ \min_{w_1 \in [0,1]} H \left( t, M_1, M_2, w_1, w_2, \frac{\partial V(t, M_1, M_2)}{\partial M_1}, \frac{\partial V(t, M_1, M_2)}{\partial M_2} \right) \right\}, \end{aligned}$$

which reduce to

$$(13) \quad \begin{aligned} u_1(t, M_1, M_2) &= \begin{cases} 0, & e^{-\mu t} + \frac{\partial V(t, M_1, M_2)}{\partial M_1} < 0, \\ 1, & e^{-\mu t} + \frac{\partial V(t, M_1, M_2)}{\partial M_1} > 0, \\ \text{arbitrary from } [0, 1], & e^{-\mu t} + \frac{\partial V(t, M_1, M_2)}{\partial M_1} = 0, \end{cases} \\ u_2(t, M_1, M_2) &= \begin{cases} 0, & e^{-\mu t} - \frac{\partial V(t, M_1, M_2)}{\partial M_2} < 0, \\ 1, & e^{-\mu t} - \frac{\partial V(t, M_1, M_2)}{\partial M_2} > 0, \\ \text{arbitrary from } [0, 1], & e^{-\mu t} - \frac{\partial V(t, M_1, M_2)}{\partial M_2} = 0. \end{cases} \end{aligned}$$

These are resource allocation strategies and functions that describe the equilibrium point and the cohorts coexisting. In other words, these equations describe the points where the two fungi enter a stalemate dynamically as time progresses.

## 5. NUMERICAL SOLUTION

As described previously, these equations are complicated to solve exactly. luckily, we do not necessarily need an exact answer. Numerical analysis is the math behind finding such answers from challenging problems. In our case, there is already a program to solve Hamilton–Jacobi equations called ROC-HJ [4]. It is written in C++ and allows for a fast and easy transcription of the functions above into machine code.

Most of the preconfigured files and procedures of ROC-HJ allow for a quick transcription of a Hamilton–Jacobi equation. [4] provides more information about how to install ROC-HJ, including further documentation. To use ROC-HJ, we must assign a few variables to describe the kind of problem. Fig. 4 contains some of the variables which describe the problem. We also need to specify both a cost function (fig. 5) and a dynamic equation (fig. 6) to produce the data that we need.

Within the `data_*.h` file, there is much more information describing the function. One can find more information and details about this project's configuration variables by directly viewing the data file [11].

In converting these equations into machine code, we must consider the Cauchy problem (12) for the HJI PDE with the nutrient fluxes (6)–(8), decay rate (9), and parameter values (10). We must further set the software to finding the solutions using a finite difference method [5, 6, 9, 13] using the ROC-HJ software [4]. One can find these “default” parameters in figure 4.

For practical purposes with ROC-HJ, it is reasonable to rewrite (12) in reverse time  $\tau = T - t$  as

$$(14) \quad \begin{cases} \frac{\partial V(T - \tau, M_1, M_2)}{\partial \tau} + \max_{u_1 \in [0,1]} \min_{u_2 \in [0,1]} \left( -H \left( T - \tau, M_1, M_2, u_1, u_2, \right. \right. \\ \left. \left. \frac{\partial V(T - \tau, M_1, M_2)}{\partial M_1}, \frac{\partial V(T - \tau, M_1, M_2)}{\partial M_2} \right) \right) = 0, \\ V(T - \tau, M_1, M_2) |_{\tau=0} = 0, \\ \tau \in [0, T], \quad (M_1, M_2) \in G, \end{cases}$$

and then to rewrite (14) in the new normalized state variables

$$m_i = M_i \cdot 10^{-4}, \quad i = 1, 2$$

(such changes of the time and state variables lead to an equivalent problem).

We specify the carrying capacity estimates as  $\bar{M}_1 = \bar{M}_2 = \bar{M} = 3.5 \cdot 10^4$ ,  $\bar{m} = \bar{M} \cdot 10^{-4} = 3.5 \cdot 10^{-4}$ , and discretize the computational region (how much the computer will increase its measurements after each pass)

$$\{(m_1, m_2, \tau) : 0 \leq m_1 \leq \bar{m}, 0 \leq m_2 \leq \bar{m}, 0 \leq \tau \leq T\}$$

by the spatial step  $\Delta m_1 = \Delta m_2 = \Delta m = 2 \cdot 10^{-3}$  and time step  $\Delta \tau = 5 \cdot 10^{-3}$ .

Monotone [B.12] finite difference methods, where there is a strictly increasing or decreasing behavior. These methods ensure theoretical convergence properties (properties where an increase in computational expenditure increases precision) with error estimates. However, they are mostly of the first order. They may generate so-called numerical diffusion problems [B.13] on long time intervals. One way to reduce and eliminate such practical issues is to use higher-order ENO (essentially non-oscillatory) schemes such as those initially proposed [13]. Higher-order ENO schemes also have been shown to have a more significant amount of stability, which is useful in eliminating practical issues. They are not necessarily monotone. There is a lack of associated convergence theorems, meaning it is challenging to ensure that the computation will approach a steady answer. However, their relevance is justifiable by various numerical experiments [5, 13].

We have used the second-order ENO scheme in ROC-HJ (fig. 4) to approximate the partial derivatives concerning the state variables, coupled with the second-order Runge–Kutta time discretization scheme. Runge–Kutta provides increased accuracy over other methods of discretely partitioning time intervals [4]. To evaluate the partial derivatives  $\partial V / \partial M_i$ ,  $i = 1, 2$ , in the saddle feedback control representations (13), we have applied the standard second-order symmetrized approximation [10, §5.7].

Figs. 1 and 2 show the reductions of the approximate value function and corresponding saddle feedback control strategies to the coordinate plane  $(m_1, m_2)$  for the time instants

$\tau = 0$  ( $t = T = 60$ ),  $\tau = 5$  ( $t = T - 5 = 55$ ),  $\tau = \tau^{**} \approx 11.175$  ( $t = T - \tau^{**} \approx 48.825$ ), and  $\tau = T = 60$  ( $t = 0$ ).

Figure 1d shows the “beginning” of the simulation, where competitive strategies have not developed. There is plenty of space and resources for both of the fungi. However, sharper edges appear as the fungi begin to either produce mycelia or produce spores. After the initial production, there is a leveling off as the fungi begin conducting spore and mycelia growth simultaneously.

Figure 1c shows that at the time point of  $t = 48.825$ , there is more differentiation in the fungi’s actions where the fungi try and compete, as there is an attempt to try and stay in the battle. Figure 1b shows that the competition increases much more near the end of the simulation. The slope becomes more pronounced as the competitive forces begin to increase.

Figure 1a shows the end of the simulation, where  $t = 60$ . The graph is flat as there are no more resources that the fungi can expend (as it is the end of the simulation). If the simulation were allowed to continue, it would continue to grow, and the slope would most likely be more pronounced than in 1b.

In figure 1, there is no clearly defined winner. This model focuses on the efforts of the fungi attempting to remain at equilibrium. However, if the resident does not attempt to remain at equilibrium, it will be invaded and eliminated.

Fig. 2 illustrates the appearance and time evolution of four approximate switching curves. For  $\tau \geq \tau^{**}$ , they intersect at the point  $(m^{**}, m^{**}) = 10^{-4} \cdot (M^{**}, M^{**})$ . With the further increase of  $\tau$ , the feedback control approaches a stationary form. There is less competition for resources between the fungi. Fig. 3 is a three-dimensional graph of the four control switching surfaces (the fungi’s direction to progress and avoid invasion). For  $\tau \geq \tau^{**}$ , the point  $(m^{**}, m^{**})$  attracts forward-time trajectories. It determines a steady-state equilibrium strategy for both cohorts. Simultaneously, one can interpret the four surfaces as necessary actions of the cohorts to avoid invasion.

From Fig. 1, one can see that, in the considered domain of initial states, the value  $V(0, M_1^0, M_2^0)$  of the zero-sum feedback differential game is negative for  $M_1^0 > M_2^0$ , zero for  $M_1^0 = M_2^0$ , and positive for  $M_1^0 < M_2^0$ . The strategy of cohort 1 is unininvadable if  $M_1^0 \geq M_2^0$ . Similarly, the strategy of cohort 2 is unininvadable if  $M_1^0 \leq M_2^0$ .

## 6. CONCLUSION

This report focuses on constructing a strategy for constructing benchmark resource allocation strategies. This paper also focuses on how differing pathogens might compete and allows us the ability to compare actual infection mechanisms. We also focus solely on resource allocation equilibrium for the one-seasonal dynamics of two biotrophic fungal cohorts within a shared host plant.

Though not mentioned in this paper, it is also relevant to investigate long-seasonal dynamics where the cohorts compete longer. Other relevant situations occur when the pathogens gain the ability to evolve, and the amount and spore-producing capacity, and when efficiencies of the pathogens begin to differ. Also interesting are situations where there are two separate and different species of fungi.

One may exploit specific discrete rules to transition from one season to another [2,12]. Another important research direction is to characterize such equilibria themselves and whether

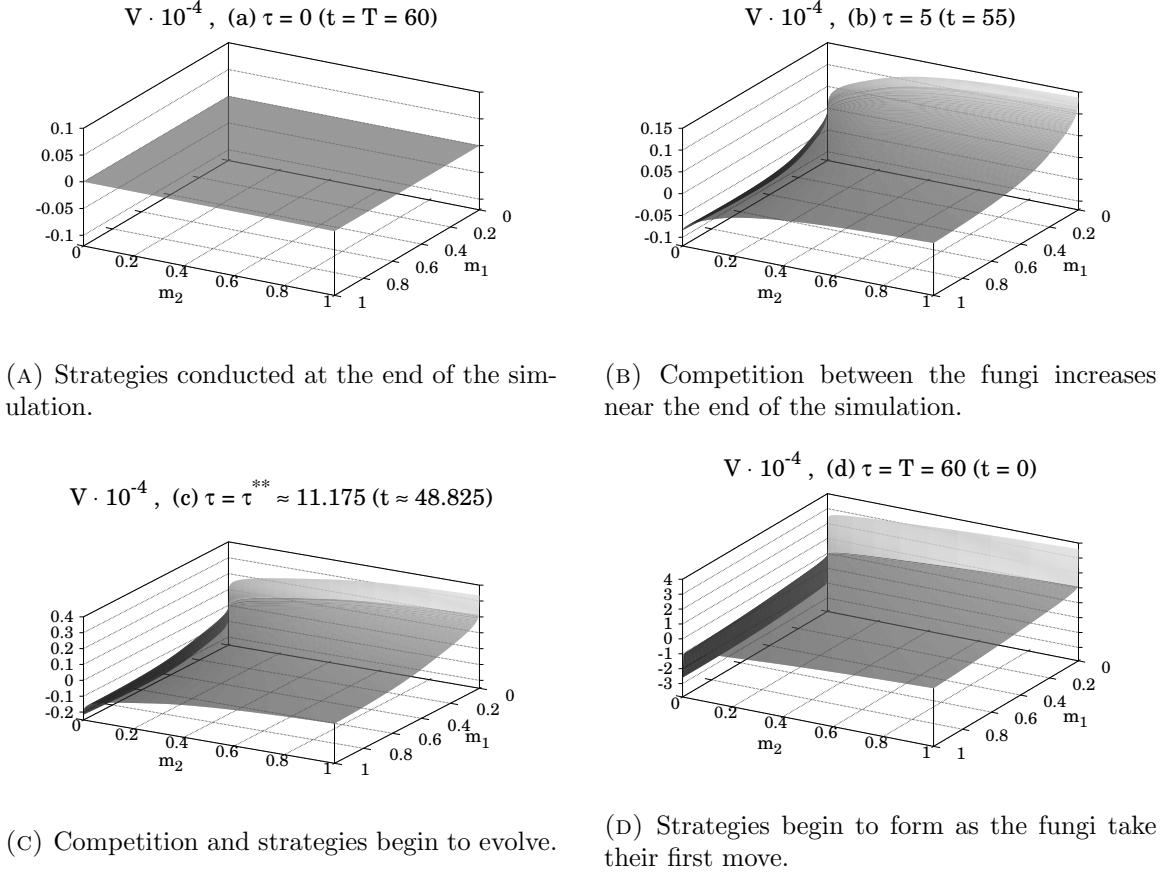


FIGURE 1. *The time instants are in reverse order to show how the computer computes the strategies using reverse time meaning the first graph is the last computation.*

they represent evolutionary attractors or not, and how the situation may change through evolution.

#### APPENDIX A. COMMON TERMS

- (1) **Zero-sum feedback game:** a game in which one of the players wants the game to reach an open target (wants to take over, invade), while the want to avoid this target forever (prevent a takeover, defend) [7].
- (2) **Pathogen:** this is an infectious organism; this paper mostly discusses fungal invaders.
- (3) **Resident pathogen:** this is the original organism, the first pathogen.
- (4) **Mutant pathogen:** this is a strain of the same species of the organism as the resident species but has a mutation through physical separation or natural mutation or otherwise.
- (5) **Mycelia:** the “roots” of a fungus, made of extremely fine branches called hyphae.

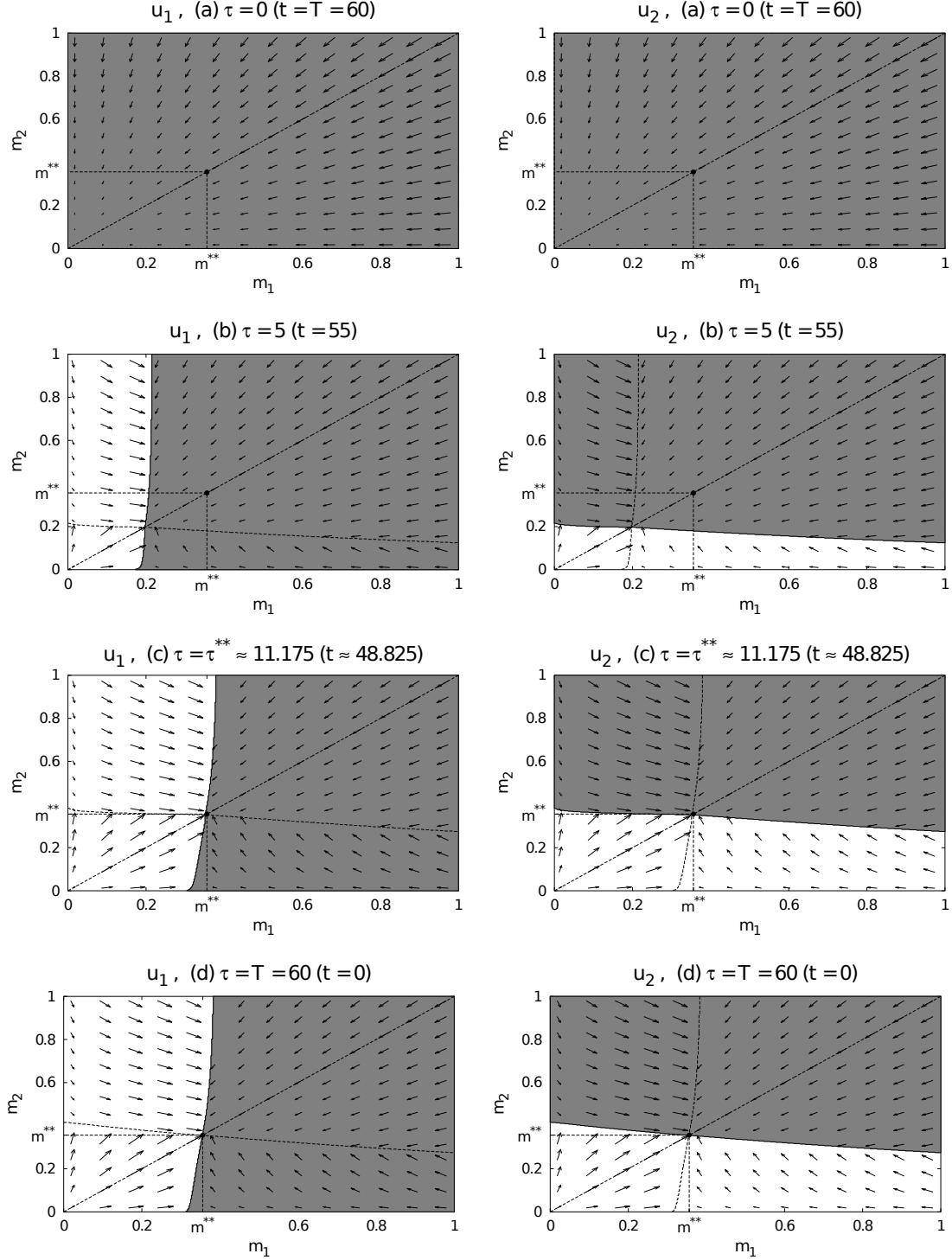


FIGURE 2. In the white regions, the fungus enters spore production. In contrast in the gray, it enters mycelia production at a particular time instance. Also, the field corresponds to its velocity of production.

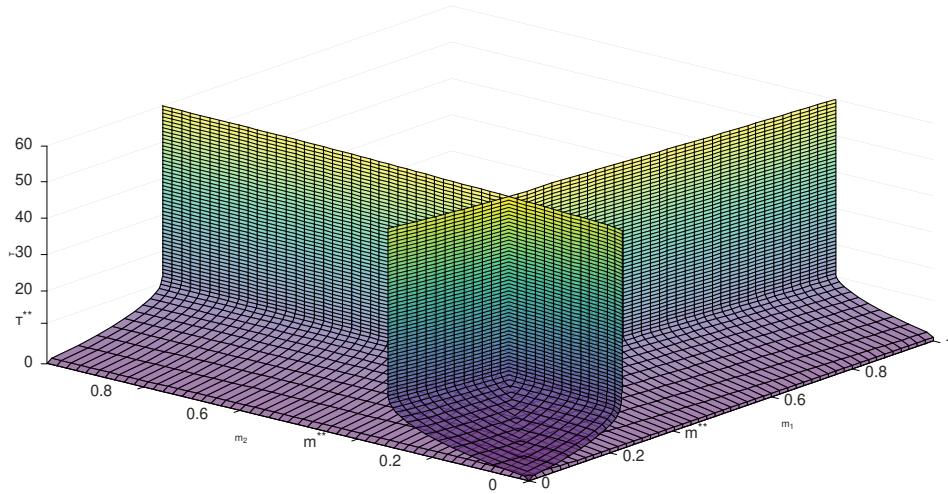


FIGURE 3. *The four control switching surfaces in the three-dimensional space  $(m_1, m_2, \tau)$ .*

- (6) **Cohort:** typically means a person or thing that is typically on “your side.” This paper typically refers to a pathogen that belongs to the same species but differs slightly and is not of the same organism.
- (7) **Biotrophic:** describes a parasite attached to a plant, a plant parasite.
- (8) **Nutrient flux:** the number of nutrients consumed by the pathogen.
- (9) **Differential game’s value:** the value of a function at a particular time when the function changes with time.
- (10) **Two marginal fitness criteria:** a value or function that is wanting to be or altered or monitored, in this document, it is typically the spore production, and the mycelia production.
- (11) **State Variables:** these are variables that set the equation or system stage and define the initial circumstances of the problem. These include items like lesion density or the rate of decay in the system.
- (12) **Saddle control strategies:** a function or system determining the saddle point of an equation.
- (13) **Carrying capacity:** the maximum size of a population that is sustainable in an environment.

#### APPENDIX B. USEFUL MATHEMATICAL DEFINITIONS

- (1) **Infimum:** the subset of the greatest elements in a set; the greatest lower bound [1].
- (2) **Supremum:** the subset of the least element that is greater than or equal to all S elements if such an element exists [1].
- (3) **Arg max/Arg min:** a set of a function in which the values are maximized (Arg max) or minimized (Arg min).
- (4) **Saddle point:** the point in which the slopes in all directions are zero, and the graph makes a saddle shape.
- (5) **Maximin / Minimax:** the largest in a series of minima (maximin) or the smallest in a maxima series (minimax).

- (6) **Cauchy problem:** a problem that asks for a partial differential equation that satisfies certain conditions given by a hypersurface.
- (7) **Non-smooth:** describes a non infinitely differentiable function.
- (8) **Viscosity solution:** a solution that is both a supersolution and a subsolution.
- (9) **Lipschitz continuous:** describing a function that has a limit on how fast it changes.
- (10) **Rademacher's theorem:** Let  $f : \mathbb{R} \rightarrow \mathbb{R}$  be any Lipschitz continuous function. Then  $f$  is differentiable in almost anywhere  $x \in \mathbb{R}^n$ .
- (11) **Lebesgue measure:** a standard method of assigning a measure to a subset, a unit of measure.
- (12) **Monotone function:** A function that preserves order, a function that is always rising or decreasing. A function is monotonically increasing if for all  $x$  and  $y$  where  $x \leq y$  then  $f(x) \leq f(y)$ , and reverse for monotonically decreasing.
- (13) **Numerical Diffusion Problems:** a difficulty that often arises when the computed medium differs from the actual medium, or it arises when it should not in the first place.

#### APPENDIX C. CODE SNIPPETS

---

```
const int OPTIM      = MAXMIN;
const int METHOD     = MFD;
const int TYPE_SCHEME = ENO2;
```

---

FIGURE 4. A few configuration variables are needed to solve the equation using ROC-HJ, MFD being the monotone finite difference method.

---

```
inline double distributed_cost2(const double* x, C u, C u2, double t) {
    double new_M_1 = x[0],
           new_M_2 = x[1],
           u_1 = u[0],
           u_2 = u2[0];
    if (new_M_1 < 0.0)
        new_M_1 = 0.0;
    if (new_M_2 < 0.0)
        new_M_2 = 0.0;
    double compet_term = 1.0 / (1.0 + new_beta * (n_1 * new_M_1 + n_2 ...
                                * new_M_2));
    return compet_term * new_alpha * exp(-mu_rate * (T - t)) *
           (u_2 * new_M_2 / (new_M_2 + new_k) - u_1 ...
            * new_M_1 / (new_M_1 + new_k));
}
```

---

FIGURE 5. Function in C++ written for ROC-HJ describing the distributed cost equation.

---

```

inline void dynamics2(const double* x, C u, C u2, double t, double* res) {
    double new_M_1 = x[0],
    new_M_2 = x[1],
    u_1 = u[0],
    u_2 = u2[0];
    if (new_M_1 < 0.0)
        new_M_1 = 0.0;
    if (new_M_2 < 0.0)
        new_M_2 = 0.0;
    double compet_term = 1.0 / (1.0 + new_beta * (n_1 * new_M_1...
        + n_2 * new_M_2));
    res[0] = new_M_1 * ((1.0 - u_1) * compet_term * new_alpha / ...
        (new_M_1 + new_k) - gamma_rate);
    res[1] = new_M_2 * ((1.0 - u_2) * compet_term * new_alpha / ...
        (new_M_2 + new_k) - gamma_rate);
}

```

---

FIGURE 6. A C++ function for use by ROC-HJ to describe the required dynamic equation.

## APPENDIX D.

### REFERENCES

1. *Chapter 1 The Real and Complex Number Systems*, chapter Chapter 1, page 4. 3 edition, 1976.
2. Andrei Akhmetzhanov, Frederic Grognaud, Ludovic Mailleret, and Pierre Bernhard. Join forces or cheat: Evolutionary analysis of a consumer–resource system. *Annals of the International Society of Dynamic Games*, 06 2010.
3. Pierre Bernhard, Frederic Grognaud, Ludovic Mailleret, and Andrei Akhmetzhanov. Ess for life history of cooperating consumers facing cheating mutants. 06 2010.
4. Desilles A, Zidani H, Bokanowski O, and J. Zhao. User’s guide for the roc-hj solver: Finite differences and semi-lagrangian methods. <https://uma.ensta-paristech.fr/soft/ROC-HJ>, 2019-01-21.
5. Olivier Bokanowski, Nicolas Forcadel, and Hasnaa Zidani. Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. *SIAM Journal on Control and Optimization*, 48, 01 2010.
6. Nikolai Botkin, Karl-Heinz Hoffmann, and Varvara Turova. Stable numerical schemes for solving hamilton–jacobi–bellman–isaacs equations. *SIAM J. Scientific Computing*, 33:992–1007, 01 2011.
7. Rainer C. Cardaliaguet P. *Zero-Sum Differential Games.*, pages 373–430. Springer, Cham, 02 June 2018.
8. Fabio Dercole and Sergio Rinaldi. *Analysis of Evolutionary Processes: The Adaptive Dynamics Approach and Its Applications*. Princeton University Press, Princeton, 11 Feb. 2008.
9. H. Fleming, W. and M. Soner, H. *Controlled Markov Processes and Viscosity Solutions*. Springer-Verlag New York, 2006.
10. Peter Frolkovič. Numerical recipes: The art of scientific computing. *Acta Applicandae Mathematica*, 19, 06 1990.
11. Yegorov I. Glasford, S. **data\_\*.h** used by roc-hj to set the problem statement.
12. Ludovic Mailleret and Valérie Lemesle. A note on semi-discrete modelling in the life sciences. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1908):4779–4799, 2009.
13. Stanley Osher and Chi-Wang Shu. High-order essentially nonoscillatory schemes for hamilton–jacobi equations. *Siam Journal on Numerical Analysis - SIAM J NUMER ANAL*, 28, 08 1991.
14. Andrei I. Subbotin. *Generalized Solutions of First-Order PDEs: The Dynamical Optimization Perspective*. Birkhäuser Basel, 1995.

15. Ivan Yegorov, Frédéric Grognard, Ludovic Mailleret, Fabien Halkett, and Pierre Bernhard. A dynamic game approach to uninvadable strategies for biotrophic pathogens. *Dynamic Games and Applications*, 10(1):257–296, Mar 2020.
16. Ivan Yegorov, Frédéric Grognard, Ludovic Mailleret, and Fabien Halkett. Optimal resource allocation for biotrophic plant pathogens. *IFAC-PapersOnLine*, 50:3154–3159, July 2017.

*Email address:* steven.glasford@ndsu.edu

DEPARTMENT OF MATHEMATICS, NORTH DAKOTA STATE UNIVERSITY, PO Box 6050, FARGO, ND 58108-6050, USA