Lagrangian formulation to optimize interactions in virtual environments

Andrea Afify [*]
Dipartimento di Fisica, Università degli Studi di Milano, Via Celoria 16, 20133 Milano, Italy

Alessandro Vicini [†]
Dipartimento di Fisica, Università degli Studi di Milano and INFN–Sezione di Milano, Via Celoria 16, 20133 Milano, Italy

Andrea Bellacicca [‡]
Compai, Milan, Italy

(Received 28 July 2025; accepted 5 January 2026; published 16 January 2026)

We study the dynamics of user-agent interactions in virtual environments through mathematical models derived in a Lagrangian formalism. The goal is to optimize the number of agents required to solve a generic predefined task while minimizing the computational cost to achieve it. Specifically, we derived Euler-Lagrange equations of motion to describe the different ways in which agents may interact with users. The main advantage of this approach lies in its peculiarity to derive a scaling law for any interaction rule one wishes to design, ensuring that the system scales while optimizing the global impact of the interactions. By specifying an action functional over interaction trajectories and enforcing its minimization, the optimal way to allocate agents emerges naturally as Euler–Lagrange equations, embedding the entire optimization strategy directly into the formalism. This intrinsic coupling of dynamics and cost yields a universal procedure to compute the scaling laws for arbitrary interaction rules, ensuring that agent populations and computational expense grow according with environmental complexity. We illustrate the methodology with two specific example. These results also offer a foundation for further research into multi-agent dynamics and their applications in virtual environments.

DOI: 10.1103/b2t1-hrmx

I. INTRODUCTION

Virtual environments have rapidly evolved as platforms for interaction, entertainment, and commerce. As virtual environments and Web3 technologies develop at an increasingly rapid pace, the ability to deploy agents capable of effectively responding to user needs has become both a critical challenge and a strategic advantage for those seeking to capitalize on the growing user base and associated market opportunities. Consequently, the design of efficient and adaptive systems for managing user-agent interactions has become an important research topic in this domain. These interactions often require robust and scalable solutions that can dynamically adapt to fluctuating user populations and changing environments, while minimizing the computational costs.

Dynamical systems theory has long been leveraged in modeling the interactions and collective behavior of multi-agent or particle systems. In the literature, this framework is often used to describe emergent phenomena such as flocking, swarming, and synchronization. For instance, agent-based models exploiting differential equations have successfully replicated patterns observed in nature, such as the alignment of bird flocks or the coordinated motion of particle ensembles (see e.g., Refs. [1–4]).

Early work, such as Ref. [5] introduced a simple yet powerful model to study the emergence of self-ordered motion in systems of particles governed by biologically inspired interactions. The model demonstrates a continuous kinetic phase transition, where particles exhibit a spontaneous symmetry breaking of rotational motion as noise levels vary, resulting in a shift from disordered to coherent motion. This approach highlights analogies between non-equilibrium systems and equilibrium models, particularly ferromagnetic systems, while incorporating dynamical properties intrinsic to transport-related phenomena.

Building on these ideas, dynamical systems have also been applied to understand predator-prey dynamics in swarming systems. In Ref. [6] a minimal model of predator-swarm interactions was developed, revealing how simple interaction laws can lead to highly organized and adaptive collective behavior. Similarly, studies such as Refs. [7] and [8] explored evasion strategies in prey swarms, illustrating the interplay between individual actions and group-level patterns that optimize survival against predation.

For animals that forage or travel in groups, decision-making often depends on social interactions among group members. Using a simple model, Ref. [9] demonstrated that informed individuals could effectively guide groups without

*Contact author: andrea.afify@unimi.it
†Contact author: alessandro.vicini@unimi.it
‡https://www.compai.team

explicit signaling, even when group members lack knowledge about who holds pertinent information. This work has inspired strategies for optimizing agent coordination in artificial systems such as Ref. [10].

In addition to natural systems, mathematical frameworks have been extended to multi-agent systems in technological and artificial contexts. For example, the limiting behaviors of multi-agent systems through convex combination updates were examined in Ref. [11], providing insights into convergence and coordination dynamics. In Ref. [12] emergent flocking behaviors in multi-agent systems using reinforcement learning were investigated, demonstrating how agents can self-organize and adapt to dynamical environments by optimizing interaction rules. A further contribution is Ref. [13] which introduce nonparametric methods to infer interaction laws directly from trajectory data, enabling the reverse-engineering of complex systems with minimal prior assumptions. Works such as Ref. [14] on scale-free networks highlight the importance of scaling laws in understanding the organization and resilience of complex systems, which are crucial for addressing scalability in multi-agent environments. In Ref. [15] the importance of viewing agent-environment interactions through the lens of dynamical systems was emphasized, illustrating how this perspective provides a unified framework for analyzing behaviors emerging from the coupling between agents and their surroundings. These studies align with broader work on emergent behaviors in human-designed systems, such as Ref. [16], in which a social force model for pedestrian dynamics was introduced, demonstrating how local interaction rules can lead to macroscopic patterns in collective movement. Our system design draws inspiration also from coevolutionary models such as those described in Ref. [17], which examined the dynamics of cyclic Lotka-Volterra systems with three-agent interactions.

While such studies focus primarily on understanding the underlying mechanics or emergent behaviors of these systems, our work introduces a different application of the dynamical systems framework. Rather than solely modeling, we leverage these mathematical tools to enhance interaction rates within virtual environments. In virtual unstructured environments, an agent tasked with interacting with users faces the challenge of adapting to unpredictable behaviors and dynamically changing conditions. A possible way to address this problem requires designing interaction frameworks not only that are responsive but whose design enable and support the system of agents to achieve predefined tasks within such complex settings. We use a Lagrangian formulation to provide the strategy governing agent-user interactions, which appears as a potential term in the Lagrangian. Our approach also leverages coupled differential equations to model these interactions, providing a rigorous and flexible tool for guiding agent behavior. Moreover, utilizing dynamical systems in this context not only facilitates task-specific adaptations but also provides scaling relationships (for example, how the required number of agents or the total computational cost grows as the problem size or the environmental complexity is changed) ensuring that the system can be expanded regardless of user-base growth or environmental properties. This dual focus, improving interaction dynamics and ensuring efficient scalability, is a significant departure from traditional applications, as it enables us to address agent modeling and operational optimization in a unified manner. Additionally, we draw upon the framework proposed in Ref. [18], which introduces positive feedback mechanisms and power-law scaling to model self-organization in dynamical systems, offering insights into how such principles can be applied to enhance agent adaptability and efficiency in complex environments.

Despite its apparent simplicity, this framework serves as a robust and versatile tool for modeling user-agent interactions within virtual environments. Beyond its theoretical roots, the framework yields significant practical advantages, notably the ability to derive agent scaling laws analytically. These scaling laws are instrumental in optimizing resource allocation, enhancing user engagement, and ensuring the robustness of interactions across dynamically varying agent populations.

The paper is structured as follows. In Sec. II we present a theoretical framework to allocate agents who have to accomplish a predefined task in a virtual environments interacting with users. In Sec. III two examples of tasks are introduced, along with their corresponding mathematical formulations using differential equations. In Sec. IV, we present numerical simulations conducted for one of the sample strategies within simplified environments. These simulations are designed to validate the theoretical results and demonstrate the practical effectiveness of our approach. Finally, Sec. V provides a summary of our results.

II. LAGRANGIAN OPTIMIZATION FOR AGENT ALLOCATION

In this section, we present a theoretical framework based on Lagrangian mechanics to model the dynamics of agent allocation considering interactions with users and a task which has to be performed. Specifically, we aim to develop a framework that rigorously prescribes how to allocate a number of agents a in a virtual environment to perform a task, optimizing interactions with a variable number of users u. The agents are influenced by the evolving state of the users, and their behaviors are coupled in a dynamic and adaptive manner. The central idea is to view the system as a coupled process, where the agent allocation is driven by a force f, influenced by the user state variable u; the user dynamics in turn evolves based on a process out of our control. We can only make minimal assumptions on the user evolution and model user dynamics based on such hypotheses. The Lagrangian formalism has been adapted to frame a numerical optimization problem rather than a physical dynamics scenario. The incorporation of the strategy $f(u)$ in the potential term, which controls the allocation of agents, allows to derive a set of equations describing the dynamics of the system for an arbitrary choice of the function f. We aim at considering a broad set of options for the function f including its implementation in an algorithmic form and we discuss case by case the time dependece of f. Finally, once also user dynamics is devised and modeled as an external source term, the coupled system is described by a set of differential equations, whose equilibrium point shifts depending on the user behavior. This model provides a mathematical framework for exploring how the agents should adapt to the changing conditions imposed by the users. By analyzing the solutions of the differential equations, i.e. the stability of the system, we can derive strategies that maximize the effectiveness of the agent-user interactions.

The proposed Lagrangian function reads:

$$\mathcal{L}(a, \dot{a}, u, t) = \left(\tfrac{1}{2}\dot{a}^2 - \tfrac{1}{2}k(a - f(u))^2\right)e^{\gamma t}, \tag{1}$$

This model is characterized by the invariance under two distinct transformations:

(1) *Translational invariance in the agent allocation coordinate.* The Lagrangian is invariant under the joint transformation $a \to a + c$ and $f(u) \to f(u) + c$. Note that adding a constant offset to both the generalized coordinate and the external input leaves the action unchanged, so only the variations of the forcing actually drive the motion. Consequently, the response is generated purely by the time-dependent variation of f.

(2) *Invariance under time shifts.* The Lagrangian is invariant under the infinitesimal transformation: $t \mapsto t + \varepsilon, a \mapsto ae^{-(\gamma/2)\varepsilon}$. We want to remark that this invariance holds considering that $f(u)$ does not transform in the same way as a, but it evolves on a timescale much longer than $\frac{1}{\gamma}$.

According to Noether's theorem, each continuous symmetry of the action gives rise to a conserved charge. For the time-translation symmetry the infinitesimal variation is $\delta a = -\frac{\gamma}{2}a$. Defining the generalized momentum $p_a = \frac{\partial \mathcal{L}}{\partial \dot{a}} = \dot{a}e^{\gamma t}$, the standard Noether charge for a transformation $(\delta t, \delta a)$, under which $\mathcal{L}$ is strictly invariant, is $J = p_a\delta a + \mathcal{L}\delta t$. Substituting the above expressions yields:

$$J = e^{\gamma t}\left[\frac{1}{2}\dot{a}^2 - \frac{1}{2}k[a - f(u)]^2 - \frac{\gamma}{2}a\dot{a}\right] \tag{2}$$

In this context, J reflects the effective work performed by the system under the effect of a general function f. The ability of the system to perform a certain amount of work is independent from the delay between the external input and the system response. Furthermore, in the limit of the undriven, undamped dynamics, J would be the usual mechanical energy which is conserved. However, in the present driven, damped dynamics, $J(t)$ provides a continuously updated measure of the cumulative work performed by the system. Its time derivative $\dot{J}$ directly quantifies the instantaneous balance between energy injected by the time-varying target $f(u(t))$ and the energy not transformed into work due to overhead. In practical terms, $J(t)$ can be interpreted as an effective performance functional of the system that quantifies how efficiently the agent allocation dynamics convert external driving signals into task execution. Because $J(t)$ aggregates the net energetic exchange between the system and its environment, monitoring its temporal evolution allows to identify operating regimes where the agents collectively adapt most efficiently to changing goals or perturbations. In multi-agent implementations, $J(t)$ can therefore serve as a feedback observable. This makes it not only a theoretical quantity associated with time-translation symmetry, but also a practically exploitable measure for adaptive control and energy-aware coordination in agent-based systems.

The dynamics of the agent–user system is obtained by demanding that the action defined as:

$$S = \int_{t_1}^{t_2}\left[\frac{1}{2}\dot{a}^2(t) - \frac{1}{2}k(a(t) - f(u(t)))^2\right]e^{\gamma t}\mathrm{d}t \tag{3}$$

is stationary under arbitrary infinitesimal variations δa that vanish at the endpoints t_1 and t_2:

$$\delta S = \int_{t_1}^{t_2}\left[\frac{\partial \mathcal{L}}{\partial a} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{a}}\right)\right]\delta a\mathrm{d}t + \left[\frac{\partial \mathcal{L}}{\partial \dot{a}}\delta a\right]_{t_1}^{t_2} = 0 \tag{4}$$

The action can be viewed as an aggregate cost functional whose minimization yields the optimal allocation trajectory $a(t)$. The kinetic term represents the resistance of the system to rapid changes in agent allocation. A more inert system resists rapid adjustments in a, leading to smaller $\dot{a}$ for the same driving force. The inertia of the system is therefore implicit in how quickly $\dot{a}$ can vary in practice. Operationally, this inertia corresponds to the average startup latency of the infrastructure, that is, the characteristic time required to instantiate agents. A system with higher latency exhibits greater inertia, reacting more gradually to fluctuations in demand, whereas a low-latency infrastructure allows faster adaptation. The potential term defines an energy landscape whose minimum corresponds to the optimal configuration $a = f(u)$. This represents the efficiency loss associated with suboptimal allocation. The constant k modulates the responsiveness of the system toward the optimum, encoding how strongly the system attempts to correct deviations from $f(u)$. In practice, k can be set to reflect a trade-off between the best achievable efficiency and cost constraints: a higher k corresponds to more aggressive corrections, while a lower k reflects deliberate moderation in response speed, ensuring that adjustments remain feasible given resource limits or operational costs. This parameter expresses the proportion of agents that the infrastructure can effectively allocate at a given time with respect to the total number theoretically required to satisfy current demand of the system. A value of $k = 1$ indicates full allocation capacity, while $k = 0.5$, for instance, denotes that only half of the necessary agents can be instantiated, reflecting partial fulfillment of the system's requirements. The exponential factor introduces a damping mechanism, which modulates the contribution of past states to the present configuration. This reflects temporal discounting, meaning that the influence of earlier fluctuations progressively decreases as the system stabilizes. In physical terms, this represents the rate at which the system dissipates "behavioral energy"–the transient mismatch between the state of the agents and the evolving user distribution. The damping parameter γ is directly related to user turnover. In practice, γ can be numerically estimated from data by fitting the exponential decay of deviations following user fluctuations or, equivalently, by associating it with the inverse of the mean user turnover rate, offering an empirically grounded calibration of the system's adaptive dynamics.

The action S quantifies the work expended by the system up to time t, while J measures the instantaneous generalized energy available (or being dissipated) at the same instant. Hence, S and J provide a tool to monitor both accumulated effort and real-time performance under varying drive and damping.

Imposing $\delta S = 0$ thus balances these competing effects to produce the damped-oscillator equation of motion that governs the coupled agent–user dynamics retrived via Euler-Lagrange equations:

$$\ddot{a} + \gamma \dot{a} = k[f(u) - a] \tag{5}$$

where γ is the damping coefficient representing the overhead of the system, k is a parameter that quantifies the efficiency with which that overhead can be reduced, and $f(u)$ is a function of the user state variable u that determines the instantaneous equilibrium value of the agents. The term $\ddot{a}$ represents the inertial contribution, while the damping term $\gamma \dot{a}$ quantifies the dissipative forces acting on the agents as they adjust to changes in the user environment.

The dynamics is characterized by the dimensionless damping ratio:

$$\zeta = \frac{\gamma}{2\sqrt{k}}. \tag{6}$$

Based on the value of ζ we can obtain 4 different regimes:

(1) undamped regime ($\zeta = 0$), (2) underdamped regime ($0 < \zeta < 1$), (3) critically damped regime ($\zeta = 1$), (4) overdamped regime ($\zeta > 1$). The solution for the undamped case is

$$\begin{aligned} a(t) &= A\cos(\omega_0 t) + B\sin(\omega_0 t) \\ &\quad + \frac{1}{\omega_0}\int_0^t \sin[\omega_0(t-s)] f(u(s))ds, \end{aligned} \tag{7}$$

In the underdamped regime we have that the solution of the corresponding equation of motion is:

$$\begin{aligned} a(t) &= e^{-\frac{\gamma}{2}t}[A\cos(\omega_d t) + B\sin(\omega_d t)] \\ &\quad + \frac{1}{\omega_d}\int_0^t e^{-\frac{\gamma}{2}(t-s)} \sin[\omega_d(t-s)] f(u(s))ds. \end{aligned} \tag{8}$$

with $\omega_d = \sqrt{k - (\frac{\gamma}{2})^2}$. For the critically damped regime the solution of the corresponding equation of motion is

$$a(t) = (A + Bt)e^{-\sqrt{k}t} + \int_0^t (t-s)e^{-\sqrt{k}(t-s)} f(u(s))ds. \tag{9}$$

Every back-and-forth swing in agent levels in response to changing user demands introduces extra cost. However, the overhead in the system dissipates the energy that would drive oscillations. In many practical systems this dissipation is sufficiently strong that inertial effects are negligible. This regime is characterized by the dimensionless damping ratio $\zeta > 1$ indicating an overdamped system. When the damping is dominant, the timescale of the inertial decay is much shorter than that of the overall evolution of $a(t)$. Under these conditions the inertial term $\ddot{a}$ may be dropped, yielding the first-order relaxation equation

$$\gamma \dot{a} = k[f(u) - a], \tag{10}$$

III. ADAPTIVE APPROACHES FOR AGENT ALLOCATION IN VIRTUAL ENVIRONMENTS

Having a framework to allocate agents that efficiently interact with users to accomplish predefined tasks is essential, because as virtual environments become increasingly complex and rich with opportunities, it gives all stakeholders a competitive advantage in reaching an ever-expanding audience. The dynamic nature of these environments—where users continuously join, interact, and depart—requires adaptive approaches to ensure that agents can respond to user needs in real time. By optimizing the allocation and behavior of agents, it becomes possible to balance resource utilization and maintain seamless interactions across different scenarios.

In this framework, predefined allocation tasks are encoded directly in the function $f(u)$, which maps the user population to the desired number of agents. By selecting $f(u)$ appropriately, it is possible to implement a broad spectrum of task objectives, such as maintaining fixed agent-to-user ratios, maximizing coverage in high-density regions, or prioritizing specific user segments. We demonstrate that for continuous and limited functions $f(u)$ the system is input-to-state stable (ISS) with respect to the unknown, time-varying user population. In practical terms, input-to-state stability guarantees that, regardless of the precise evolution of $u(t)$, the allocation of agents will remain bounded and will asymptotically track the target up to a margin proportional to the rate of change of the user input. This property provides a robust foundation for implementing agent allocation heuristics in real virtual environments, where user behavior cannot be predicted *a priori*, while ensuring that agents continuously and smoothly adjust to fulfill the encoded task objectives.

Furthermore, this section shows two examples: the "*Assistant Strategy*", which focuses on maintaining an optimal agent-to-user ratio through dynamical adjustment, and the "*Seller Strategy*", which emphasizes the maximization of the interactions between agents and users. Depending on the objective and the context, it is possible to decide which of the two strategies is more appropriate to adopt. For example, let's consider a virtual art gallery where users are guided by virtual agents. Each agent acts as a guide, providing personalized explanations and responding to questions from the users in their assigned group. To ensure a high-quality experience, a constant agent-to-user ratio must be maintained, so that for a fixed number of users, there is always a dedicated guide. As the number of users increases, additional guides are dynamically created to maintain this ratio. This dynamical adjustment ensures that each user receives adequate attention without overloading agents or wasting computational resources. By leveraging the assistant strategy, the system achieves scalability and maintains consistent interaction quality, critical for delivering an immersive art gallery experience. In contrast, let's consider a virtual shopping center where the goal is to maximize the interactions of users with agents that represent various brands or stores. Here, the primary focus is not on providing consistent assistance but on maximizing the probability of agent-user interactions to promote products and drive sales. The Seller Strategy is applied by deploying agents in locations with the highest user density, such as entrances, popular store areas, or event zones. The system minimizes the computational cost by strategically limiting the number of agents to those necessary for effective coverage while ensuring a high likelihood of interactions. The differential equations governing this strategy focus on increasing the exposure of users to promotional activities.

For both of the examples below, we consider an overdamped system. Let the inertial decay time be $\tau_m = \frac{1}{\gamma}$ and denote by τ_{u} the characteristic timescale over which the user-driven target $f(u(t))$ evolves. When $\tau_m \ll \tau_{\mathrm{u}}$ any transient acceleration of $a(t)$ is damped out in a time much shorter than the variations of $f(u(t))$. Hence, the equation of motion can be simplified, yielding the first-order (overdamped) law described in Eq. (10). Physically, this approximation means that when user dynamics changes slowly on the network timescale τ_{u}, the system's resistance to acceleration can be neglected, and $a(t)$ simply relaxes toward $f(u(t))$ without measurable inertial lag or oscillation.

The stability analysis carried out for both strategies revealed the existence of stable equilibrium points, which are essential to ensure the effectiveness of the analytical solutions obtained and to implement them with guaranteed stability.

A. Stability of agent allocation under unknown user dynamics

In practical virtual environments, the user population $u(t)$ is generally unknown and time-varying. We now formalize the stability of agent allocation heuristics under these realistic conditions.

Theorem 1. Consider the overdamped agent dynamics

$$\gamma \dot{a}(t) = k(f(u(t)) - a(t)), \quad k, \gamma > 0,$$

where $f : \mathbb{R}^+ \to \mathbb{R}^+$ is continuous and bounded, and $u(t) \geqslant 0$ is an unknown, possibly time-varying user population. Define the instantaneous target allocation:

$$a^*(t) = f(u(t)), \quad \delta a(t) = a(t) - a^*(t).$$

Then the system is input-to-state stable (ISS) with respect to the unknown input $u(t)$:

$$|\delta a(t)| \leqslant |\delta a(0)| e^{-(k/\gamma)t} + \frac{k}{\gamma}\int_0^t e^{-(k/\gamma)(t-s)} |\dot{a}^*(s)| ds.$$

Proof. Define the deviation $\delta a(t) = a(t) - a^*(t)$. Differentiating, we obtain:

$$\dot{\delta a}(t) = \frac{k}{\gamma}(f(u(t)) - a(t)) - \dot{a}^*(t) = -\frac{k}{\gamma}\delta a(t) - \dot{a}^*(t),$$

which is a linear first-order system with an input $-\dot{a}^*(t)$.

The solution of this linear ordinary differential equation is:

$$\delta a(t) = e^{-(k/\gamma)t}\delta a(0) - \frac{k}{\gamma}\int_0^t e^{-(k/\gamma)(t-s)} \dot{a}^*(s) ds.$$

Taking absolute values and applying the triangle inequality yields:

$$|\delta a(t)| \leqslant |\delta a(0)| e^{-(k/\gamma)t} + \frac{k}{\gamma}\int_0^t e^{-(k/\gamma)(t-s)} |\dot{a}^*(s)| ds.$$

This inequality demonstrates that the deviation $|\delta a(t)|$ is bounded in terms of the initial error and the rate of change of the unknown target $a^*(t)$, i.e., the system is input-to-state stable. ∎

The result applies to any continuous bounded $f(u)$. If $u(t)$ varies slowly, the tracking error $|\delta a(t)|$ remains small and $a(t)$ closely follows $f(u(t))$. Rapid variations in $u(t)$ increase the transient tracking error proportionally, but the system remains stable. The overdamped Lagrangian formulation encodes a continuous family of agent allocation heuristics that are guaranteed to track unknown, time-varying user populations in a smooth and stable manner.

B. Assistant strategy

Maintaining a constant agent-to-user ratio is critical in environments where seamless interactions, equitable resource distribution, and operational balance are required. In this context, users of the virtual environment are entities seeking services, resources, or interactions. Users can connect and disconnect from the virtual environment at any time; the inflow and outflow of users over time proceeds with a certain rate α. At the same time, a certain percentage β of a agents will complete their duties, causing a decrease of the user population (u), or some users will decline the assistance offered by an agent. As a matter of fact, once a user has been served or declined a proposal from an agent, it should be removed from further consideration for similar interactions. Refusal implies that additional engagement would be unnecessary and potentially unwelcome.

Instead, agents are tasked with supporting or providing resources to users. Thus, it becomes essential to devise a dynamical system capable of ensuring that each agent consistently covers a predefined number of users. To address this, the constant $C = u/a$ represents the user-to-agent ratio required to reliably assist all users in the environment. Agents increase or decrease to ensure that the number of agents meets the required coverage. The dynamics is described by the equations:

$$\begin{aligned} \dot{u} &= \alpha u - \beta u a \\ \dot{a} &= \eta\left(\frac{u}{C} - a\right) \end{aligned} \tag{11}$$

In Eq. (11) $\eta = \frac{k}{\gamma}$ is a proportionality factor that controls how quickly the agent population a adjusts to reach the desired level based on u, while $\frac{u}{C} - a$ represents the gap between the current agent population and the required agent population to cover all users. This term drives a to the level where $a = \frac{u}{C}$.

To analyze the dynamics described by the system of differential equations, we begin by identifying the equilibrium points through

$$\begin{aligned} \alpha u - \beta u a &= 0 \\ \eta\left(\frac{u}{C} - a\right) &= 0 \end{aligned} \tag{12}$$

By solving Eq. (12), we find that the system has two equilibrium points: $(u, a) = (0, 0)$ and $(u, a) = (\frac{\alpha C}{\beta}, \frac{\alpha}{\beta})$. We start by computing the Jacobian matrix of the system:

$$J = \begin{pmatrix} \alpha - \beta a & -\beta u \\ \frac{\eta}{C} & -\eta \end{pmatrix} \tag{13}$$

Evaluating the Jacobian matrix at the first equilibrium point $(u, a) = (0, 0)$, we find that the eigenvalues coincide and are equal to $-\eta$ when $\alpha + \eta = 0$; otherwise, they are α and $-\eta$. However, $-\eta$ is an unacceptable value, and in all other cases, the equilibrium point is unstable. Then, evaluating the Jacobian matrix at the second equilibrium point $(u, a) = (\frac{\alpha C}{\beta}, \frac{\alpha}{\beta})$

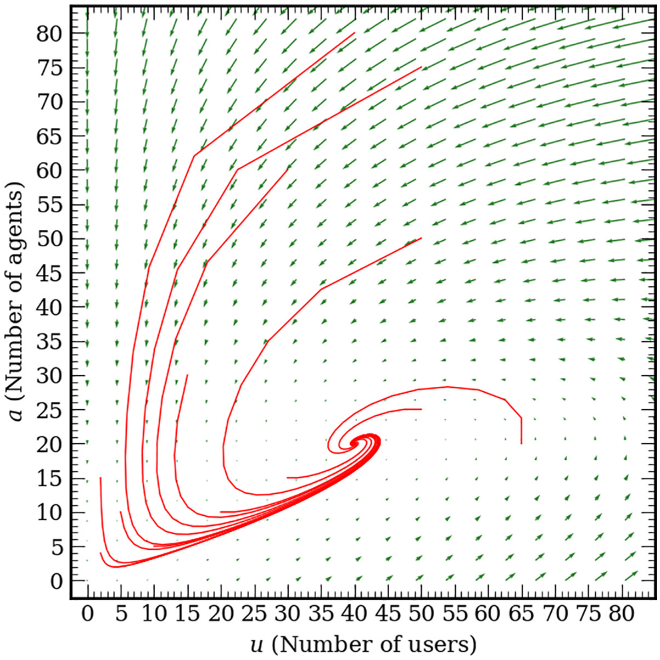

FIG. 1. The phase portrait when $\eta^2 - 4\alpha\eta < 0$ with $\eta = 0.3$, $\alpha = 0.2$, $\beta = 0.01$ and $C = 2$. The dark green arrows denote the vector field, illustrating the rate and direction of change in user and agent populations.

we find that the eigenvalues are given by:

$$\lambda_{1,2} = \frac{-\eta \pm \sqrt{\eta^2 - 4\alpha\eta}}{2} \tag{14}$$

If $\eta^2 - 4\alpha\eta \geqslant 0$, the eigenvalues are real, indicating stability (negative eigenvalues) or instability (positive eigenvalues). If $\eta^2 - 4\alpha\eta < 0$, the eigenvalues are complex conjugates with negative real parts, and $(u, a) = (\frac{\alpha C}{\beta}, \frac{\alpha}{\beta})$ is a stable equilibrium point (see Fig. 1).

C. Seller strategy

We consider the realistic case in which companies deploy agents in a virtual environment to sell their products to users as quickly as possible. In this scenario, several challenging tasks must be addressed. First of all, agents have to recognize users in the environment. Next, once a user has been detected, the agent needs an effective strategy to navigate toward them. Each agent could detect users using a real-time object detection algorithm as you only look once framework presented in Ref. [19] which identifies specific objects in videos, live feeds, or images. However, although it is a highly effective algorithm for recognizing users within the virtual environment, the challenge lies in the fact that a single agent cannot guarantee that each user is both detected and reached, even when only one user is present in the environment. This limitation results from the intrinsic limit of the object detection task (e.g. missed frames, occlusions, false negatives).

With this second strategy, we aim to ensure that, at equilibrium, the system saturates the user base, ultimately reducing to zero the number of agents involved in the activity. Let's consider p as the probability that a user interact with an agent between the a agents available. Thus, we can define $(1-p)^a$ as the probability that an user will not interact with any agent. So the probability that an user will interact at least with one agent is $1-(1-p)^a$. We want to maximize this probability as

$$1-(1-p)^a \geqslant 1-\epsilon \tag{15}$$

Solving for a we have that we need at least $a = \log_{(1-p)}(\epsilon)$ agents and the dynamics of this system can be described by

$$\begin{aligned}\dot{u} &= \alpha u - (1-(1-p)^a)u\\ \dot{a} &= \eta(\log_{(1-p)}(\epsilon)u - a)\end{aligned} \tag{16}$$

In the first differential equation the term αu represents the natural influx of users into the environment at a rate proportional to the current population u. The term $(1-(1-p)^a)u$ represents the probability that users will interact with at least one agent, given a agents with individual interaction probability p. This interaction probability is subtracted from αu, representing users who have been reached. The second differential equation describes the rate of change in the agent population a. The term $\log_{1-p}(\epsilon)u$ is a factor that determines the required agents to maximize the interaction probability based on the number of users available. Specifically, it defines the effective number of agents needed to ensure that, with probability $1-\epsilon$ each user is covered by at least one agent.

To analyze the dynamics described by the system of differential equations, we begin by identifying the equilibrium points through:

$$\begin{aligned}\alpha u - (1-(1-p)^a)u &= 0\\ \eta(\log_{1-p}(\epsilon)u - a) &= 0\end{aligned} \tag{17}$$

Solving Eq. (17) it is found out the system has two equilibrium points: $(u, a) = (0, 0)$ and $(u, a) = \left(\frac{\log_{1-p}(1-\alpha)}{\log_{1-p}(\epsilon)}, \log_{1-p}(1-\alpha)\right)$. We start by computing the Jacobian matrix of the system:

$$J = \begin{pmatrix} \alpha - (1-(1-p)^a) & au(1-p)^{a-1} \\ \eta \log_{1-p}(\epsilon) & -\eta \end{pmatrix} \tag{18}$$

Evaluating the Jacobian matrix in the first equilibrium point $(u, a) = (0, 0)$ we obtain the same results found in the previous assistant strategy. We can conclude that $(u, a) = (0, 0)$ is an unstable equilibrium point except when $\eta = -\alpha$; however in this case $-\eta$ is an unacceptable value (see Fig. 2). Then, we can evaluate the second equilibrium point $(u, a) = \left(\frac{\log_{1-p}(1-\alpha)}{\log_{1-p}(\epsilon)}, \log_{1-p}(1-\alpha)\right)$ and we find that eigenvalues are given by:

$$\lambda_{1,2} = \frac{-\eta \pm \sqrt{\eta^2 + \frac{4\eta(1-\alpha)\log^2_{1-p}(1-\alpha)}{1-p}}}{2} \tag{19}$$

IV. RESULTS AND DISCUSSION

In this section, we describe the simulation setup and the results achieved implementing the seller strategy in a bidimensional virtual environment. The need to build a simulation—even if it's conducted in a two-dimensional virtual environment that abstracts away many of the features

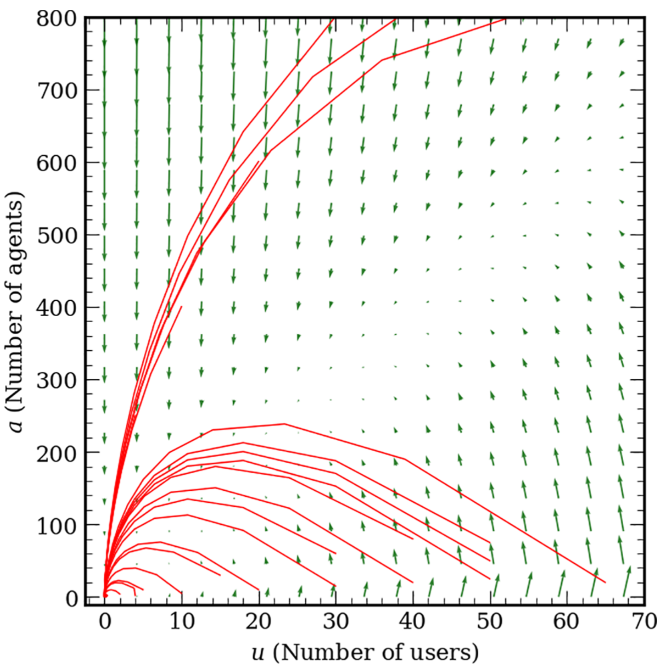

FIG. 2. The phase portrait for the seller strategy with $\eta = 0.3$, $\alpha = 0.6$, $p = 0.4$ and $\epsilon = 0.01$. The dark green arrows denote the vector field, illustrating the rate and direction of change in user and agent populations.

of its three-dimensional counterpart—responds to the necessity of validating the theoretical framework presented in the previous sections under more realistic conditions, where user dynamics are not governed by analytic equations but by a stochastic process. Moreover, this approach allows us to quantitatively compare the strategy described earlier against benchmarks defined by a random process to allocate agents and a uniform allocation. It also provides a standardized setup and performance metric for evaluating and comparing future strategies, whether they are analytical or numerical.

We conduct simulations within a simplified environment, to isolate and thoroughly evaluate the core strategies of the framework. We ensure in this way that our focus remains entirely on the behavior and performance of the strategy itself, without introducing the complexities and computational overhead inherent to a metaverse environment. While implementing and reproducing the strategy directly within a metaverse would offer a more immersive evaluation, such an approach is computationally expensive and resource-intensive. As a result, we opted to initially validate the framework in a streamlined, controlled setting that mirrors key aspects of the virtual world but excludes extraneous factors. This approach allows us to efficiently gain an initial understanding of the strategy's effectiveness and its potential for adaptation to more complex scenarios in a subsequent phase of the research.

In this context, it is important to remark that, despite the use of a two-dimensional grid, our simulation setup faithfully captures the essential locomotion and interaction features of real metaverse environments, such as The Nemesis [20] and Decentraland [21], which are among the most prominent platforms currently available. Although these environments are rendered in three dimensions, their spatial organization is fundamentally based on a discretized Cartesian grid of land parcels indexed by integer coordinates (x, y). Within each parcel, avatar positions are represented in local 3D coordinates, but user control is effectively restricted to planar navigation. Movement commands—typically forward, backward, left, and right—span only two degrees of freedom on the ground plane, while the third spatial dimension (height) emerges as a dependent variable governed by the terrain geometry or animation system rather than by user or agent decisions. Consequently, modeling agent dynamics on a 2D grid captures the critical decision-making and allocation behavior of avatars, providing a computationally efficient yet realistic setup for evaluating strategy performance.

The simulation evaluates the impact of the seller strategy on a bidimensional user-agent environment. It is conducted in a grid-based environment of dimensions 800×600 pixels, with each grid cell having a size of 20×20 pixels. Agents and users move randomly in the grid. The initial population of users and agents is set to 100 and 10 respectively. Additional users are introduced stochastically during the simulation. Specifically, at each time step, a fixed number of 10 users are added to the environment with a probability of 0.5, provided the total number of users does not exceed 3000. This stochastic process mimics random user logins into the system. The number of agents is dynamically adjusted based on the seller strategy, governed by the differential equation in Eq. (16). This equation modulates the agent population to optimize interactions with users. When a user is in the same cell as an agent, an interaction occurs. In particular, the user is removed from the simulation, either because they have accepted the offered product and should no longer be considered, or because they have declined and the agents should not persistently attempt to sell to them. When all the users have been reached and the agents have interacted with all of them the simulation is stopped. In addition to the seller strategy, we have incorporated two alternative rules for adding agents to the simulation. These rules, referred to as the random strategy and the uniform strategy, were introduced to establish benchmarks for evaluating the effectiveness of the seller strategy. The first method, known as the random strategy, involves adding a random number of agents to the simulation at each step. Specifically, the number of agents introduced is a randomly chosen value that can range from 0 to the current total number of users present in the simulation. The second method, referred to as the uniform strategy, employs a deterministic approach. In this case, at each step of the simulation, the number of agents added is exactly equal to the number of users present in the environment at that moment. This ensures a one-to-one correspondence between the number of users and agents added, providing a consistent and evenly distributed agent deployment.

To evaluate and compare these strategies, we conducted 20 separate simulation runs for each strategy. During these simulations, we recorded the number of steps required to reduce the number of users in the environment to zero. This metric serves as a measure of the strategy's efficiency in achieving user-agent interactions. The results of these experiments are summarized and presented in Fig. 3.

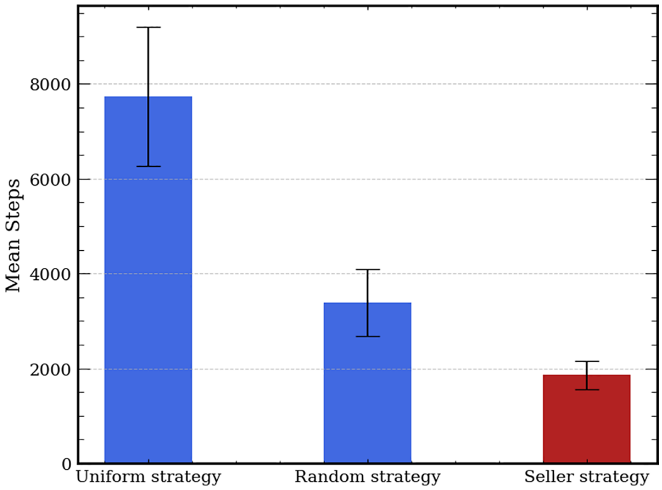

FIG. 3. Mean number of steps for each strategy to reduce users to zero are depicted with the standard error of the mean.

While the theoretical coverage dynamics [Eq. (16)] is formulated in continuous space and time, the grid-based implementation represents a straightforward discretization of that model. Each cell stands in for a small region of the continuous virtual world, and each simulation time-step corresponds to a fixed Δt. As long as Δt is chosen sufficiently small relative to the temporal scales of user–agent interactions, the discrete approximation reproduces the trajectories and population adjustments prescribed by the continuous differential equation. Consequently, the steady-state and transient performance metrics (e.g. the time to cover all users) converge to the same values that one would obtain by integrating the continuous model.

V. CONCLUSIONS

This study presents a theoretical and computational investigation into the dynamics of users and agents within a simplified virtual environment. This abstraction has demonstrated, through specific examples, how dynamical systems and a Lagrangian formulation of the problem can be exploited as a mathematical framework to achieve predefined goals in a virtual environment, with the scaling of agents governed by the dynamics of the system. Exploiting the lagrangian framework, we have optimized the allocation of agents in response to an evolving user state. By casting the allocation dynamics in Lagrangian form we observe the presence of two invariant quantities, due to translational in the allocation coordinate and time–translation symmetries. The equations of motion naturally incorporate both the reallocation effort and irreversible adaptation costs. The first invariance ensures that only deviations from the user-driven equilibrium drive the dynamics, while the second encodes the intrinsic dissipative losses encountered in real resource reallocation. The corresponding Euler–Lagrange derivation yields a damped-oscillator equation in which the instantaneous equilibrium shifts according to the user strategy. This offers the possibility to recover optimal reallocation trajectories as those minimizing an aggregate cost functional.

This approach helps to build task-specific dynamics for agents: the challenge is the achievement of a task in presence of an unpredictable behavior of the users. In many real scenarios the strategy $f(u)$ could be learned and user dynamics could vary through time. In the future, we will consider agents with a more complex behavior and able to autonomously interact with users; such developments will allow the simulations e.g. of complex systems such as those describing social groups. The increasing computational costs will require a systematical optimization, like the one discussed in the present paper. A proper theoretical framework will help the characterization of anomalous or emergent phenomena, such as those relevant in all the evolution phases of a dynamical system implemented in a metaverse.

ACKNOWLEDGMENT

The authors would like to thank M. Gherardi for insightful discussions and his valuable suggestions, which significantly contributed to the development of this work.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

[1] C. M. Topaz and A. L. Bertozzi, SIAM J. Appl. Math. **65**, 152 (2004).
[2] J. Toner, Y. Tu, and S. Ramaswamy, Ann. Phys. **318**, 170 (2005).
[3] W. Bialek, A. Cavagna, I. Giardina, T. Mora, E. Silvestri, M. Viale, and A. M. Walczak, Proc. Natl. Acad. Sci. USA **109**, 4786 (2012).
[4] G. K. Sar and D. Ghosh, Chaos **33**, 123126 (2023).
[5] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, Phys. Rev. Lett. **75**, 1226 (1995).
[6] Y. Chen and T. Kolokolnikov, J. R. Soc. Interface **11**, 20131208 (2014).
[7] M. Zheng, Y. Kashimori, O. Hoshino, K. Fujita, and T. Kambara, J. Theor. Biol. **235**, 153 (2005).
[8] J. M. Jeschke and R. Tollrian, Anim. Behav. **74**, 387 (2007).
[9] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, Nature (London) **433**, 513 (2005).
[10] G. Wang, T. V. Phan, S. Li, M. Wombacher, J. Qu, Y. Peng, G. Chen, D. I. Goldman, S. A. Levin, R. H. Austin, *et al.*, Phys. Rev. Lett. **126**, 108002 (2021).

[11] S. Bolouki and R. P. Malhamé, IFAC Proc. Vol. **44**, 8819 (2011).

[12] M. Durve, F. Peruani, and A. Celani, Phys. Rev. E **102**, 012601 (2020).

[13] F. Lu, M. Zhong, S. Tang, and M. Maggioni, Proc. Natl. Acad. Sci. USA **116**, 14424 (2019).

[14] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).

[15] R. D. Beer, Artif. Intell. **72**, 173 (1995).

[16] D. Helbing and P. Molnar, Phys. Rev. E **51**, 4282 (1995).

[17] F. Palombi, S. Ferriani, and S. Toti, Eur. Phys. J. B **93**, 194 (2020).

[18] M. Brouillet and G. Y. Georgiev, Processes **12**, 2937 (2024).

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, 2016), pp. 779–788.

[20] https://thenemesis.io/.

[21] https://decentraland.org/.