PhD programme in Computer Science

Carolina Crespi

From Ants to Crowds
Harnessing Competition for Optimization

PhD thesis

Supervisor:
Prof. Mario F. Pavone

academic year 2022/2023

Contents

Abstract

This doctoral thesis investigates competition dynamics within complex systems, particularly within the framework of network science. It addresses the interplay between cooperation and competition in various domains, such as optimization problems and social complex systems. The research explores the role of competition within cooperative frameworks, aiming to understand its significance in achieving optimal solutions. Three interconnected research lines were developed, each employing agent-based models. The first line developed an Ant Colony Optimization algorithm for dynamic networks, showing that strategic competition can enhance algorithm efficiency. The second line focused on collective behaviors, revealing that a balanced mix of cooperation and competition in crowds yields better outcomes. The third line examined trust dynamics in multi-agent systems, finding that trusting a robot, especially with higher efficiency, enhances agent performance. Overall, the research highlights the intricate dynamics of cooperation and competition in complex systems, shedding light on their combined impact on system behavior and performance.

Introduction

Observing various systems in the surrounding world reveals fundamental similarities despite their apparent differences. For instance, when considering society, its functioning results from a delicate process of interaction and optimization involving individuals and various infrastructures simultaneously. From the Earth's global climate to ecosystems, from infrastructures like the power grid to transportation or communication systems, and from social and economic organizations to the spread of epidemics. What characterizes these systems is the challenge of extrapolating models from them and understanding their evolution over time due to dependencies, relationships, or other types of interactions between their parts or between the entire system and the environment.

The study of these systems falls under Network Science, a discipline rooted in graph theory and sociology. The birth of Network Science can be attributed to two key scientific works, particularly the 1959 study by Paul Erdős and Alfréd Rényi on random networks [29] and Mark Granovetter's 1973 work on weak ties in social networks [34]. While these works were initially confined to their specific domains and remained a form of abstract mathematics, the 21st century marked the true emergence of Network Science as a field. Historically, understanding and studying the behavior of those systems, which are composed of hundreds to billions of interacting components, involved creating detailed maps of their elements and interactions. However, the technological revolution has shifted this paradigm, introducing a new approach where systems are viewed as networks. In this framework, nodes represent the system's

elements and links are the relationships between them. In this way, it has been possible to identify a common denominator among vastly different networks: the nodes of a metabolic network are molecules, and the links represent the chemical reactions between them, governed by the laws of chemistry; the nodes of the Web represent documents, and the links between them are URLs regulated by appropriate algorithms; the nodes of social networks represent individuals, and the links between them represent familial, professional, friendship, or acquaintance connections; the nodes in transportation networks represent locations (physical and otherwise), and the links represent roads or other routes [58]. In essence, this abstraction allows for examining diverse systems, using a common set of mathematical tools and a complex system to be identified as a complex network.

In network science, challenges often revolve around identifying optimal solutions from a multitude of possibilities, collectively referred to as optimization problems. These encompass a wide range of scenarios, spanning from planning strategies to combat climate change and preserve ecosystems, to optimizing traffic flow and routing, and even developing strategies to combat the spread of epidemics. For instance, in addressing Earth's global climate change, the objective is to minimize greenhouse gas emissions. This involves tasks such as determining optimal locations for installing renewable energy facilities [8] and devising cost-effective strategies for transitioning to renewable energy, considering the economic and geographic contexts of different countries [73]. Similarly, in ecosystems, the goal is to protect biodiversity while accounting for economic and social factors. Optimization techniques can help identify key conservation areas [56], manage timber harvesting to maintain forest health [44], and establish sustainable fishing quotas to prevent overexploitation while ensuring economic viability and ecosystem longevity [7]. In transportation systems, optimization problems often focus on designing efficient public transportation routes that minimize travel time for passengers [37]. Flow optimization is crucial for alleviating traffic congestion by adjusting traffic light timings and routes, ensuring smooth traffic flow within city networks. In emergency response planning, determining the fastest ambulance routes considering traffic conditions and road closures is essential [16]. Furthermore, routing problems encountered in ride-sharing

platforms and delivery services aim to match users with drivers or optimize paths while considering route efficiency, availability, and passenger preferences [50]. In the context of combating epidemics, optimization plays a vital role in developing data-driven strategies to contain outbreaks and protect populations. This includes designing effective social distancing measures that balance the need to control the spread of disease with minimizing economic and social disruption [32]. In essence, through the lens of network science, these optimization efforts contribute to enhancing the efficiency, fairness, and sustainability of various systems, thereby exerting a profound influence on multiple aspects of daily life.

Traditionally, these problems were tackled using exact methods capable of finding the optimal solution to the problem. However, applying these methods is not always the most suitable choice. Firstly, because the networks being tested have sizes that are gradually increasing over the years, making exact methods computationally expensive. Secondly, real-world systems' networks [69, 70, 36, 82] present non-trivial topological characteristics, such as a power-law degree distribution, a high clustering coefficient, evolving over time, and generally having properties and structures that are not typical of regular or random networks [6, 94]. This is why approximate methods are now preferred. Networks of this type are called dynamic networks and are studied in the realm of Network Dynamics, a relatively recent field that explores networks with evolving structures and properties.

Within Network Dynamics, approximate methods, particularly those rooted in *Swarm Intelligence* algorithms, emerged as viable alternatives. Swarm Intelligence refers to the collective behavior of decentralized, self-organized systems, where the group's intelligence surpasses that of individual members. These algorithms are thus inspired by nature and can be categorized as insect-based, bacteria-based, or bird-based, depending on their source of inspiration [10]. Their efficacy, especially in the optimization of dynamic networks, highlights the potential of nature-inspired approaches in problem-solving. Nature, particularly Darwinian evolution, serves as a rich source of inspiration for developing algorithms that adapt and optimize in complex environments. This adaptation is modeled by viewing solutions as individuals in a population, evolving based on principles of variation, adaptation, and heredity [1].

The ability of these systems, both in reality and in their algorithmic version, to arrive at a solution lies in the fact that by being composed of numerous elements interacting with each other and the environment, they give rise to emergent behaviors—macroscopic and global properties of the system that cannot be deduced from the characteristics of individual elements [26]. In detail, the result arises from the cooperation among elements of the group, where no predefined leader is coordinating the group. Cooperation is the manifestation of an emergent behavior stemming from interactions between group elements. Such systems, because of their features, are called Complex Adaptive Systems (CAS) and some examples are observed in various contexts such as climate [5], traffic flows [88], financial markets [92], and social networks [52] but also ant colonies and bee swarms [2].

A deeper analysis of these systems reveals the presence of not only cooperative dynamics but also competitive ones. For instance, insect societies, typically regarded as cohesive groups of individuals who collaborate for the collective benefit, have recently been investigated, uncovering conflicts of interest among group members [38]. The range of interactions within these societies spans from aggressive conflicts to mutual tolerance and cooperation among spatially separate groups [67].

In more complex animal species, like mammals, group members may cooperate to defend resources against neighboring groups or to detect and deter predators [41]. Nevertheless, it has been observed that female mammals engage in competition, striving for both resources and mates to secure reproductive benefits [75].

Shifting the focus to the workplace, a similar coexistence of cooperative and competitive dynamics is observed, where employees collaborate to achieve shared objectives while simultaneously competing for promotions [57]. Additionally, in sports, the simultaneous presence of cooperative and competitive elements is evident. Game sports, for instance, demand both elements as competition fuels the cooperative aspect, highlighting the intricate interplay between these dynamics [59].

In summary, the acknowledged effectiveness of cooperation in achieving better results is counterbalanced by the inherent presence of competition within the same systems, as illustrated in the examples mentioned. The recognition that cooperation and competition are inseparable

prompts a pivotal question: *What role does competition play within complex systems that also rely on cooperation?*

The pursuit of answering this question guided the unfolding of the presented doctoral project along three interconnected research lines. These lines evolved from the initial question, delving into diverse domains, with each analyzed in a distinct research line.

In the **first research line**, the primary question addressed was: *Could competition contribute to the optimization of outcomes in the context of optimization problems?* This question led to the development of a tailored Ant Colony Optimization algorithm (ACO) for dynamic networks. The algorithm was implemented through an agent-based model where cooperative and competitive ant colonies navigate virtual environments, assessing whether the introduction of competitive dynamics could enhance the efficiency of the ACO algorithm. This research line resulted in two published works: [25] and [23] and a third work, [14] about group dynamics in memory-enhanced ants.

Concurrently, the **second research line** delved into the social aspects of collective behaviors by posing the question: *What is the interplay between cooperation and competition in social complex systems?* This inquiry not only prompted the need for a social model but also led to a subsequent research question: *Can the principles of the ACO algorithm be applied to simulate social dynamics?* Through a process of abstraction, a coherent mapping of the ACO algorithm's rules into conceptual frameworks rooted in social dynamics was proposed. Subsequently, an agent-based model was implemented to analyze collective dynamics within crowds, with various investigations conducted to comprehend the impacts of distinct behavioral strategies, particularly focusing on cooperation and defection, within crowd contexts. This research line resulted in four published works, [21], [20], [19], [24] and one in press [22].

The **third research line**, undertaken in collaboration with Professor Angelo Cangelosi from the *University of Manchester* and Marta Romeo, Assistant Professor in the Computer Science department at *Heriot-Watt University*, introduced a distinct perspective. Leveraging the agent-based model developed in the second research line for collective behaviors, an adaptation was implemented to simulate social behaviors associated with trust. This adjustment aimed to align the research with the research in-

terests of Professors Cangelosi and Romeo, leading to a shift in focus towards a different research question: *What are the effects of trust dynamics in multi-agent systems?* In addressing this question, the model was used to comprehend the impacts of distinct behavioral strategies, here with a specific emphasis on trust and skepticism, particularly in the context of a robot assisting in navigation through unfamiliar environments. This research line resulted in a submitted work to The Genetic and Evolutionary Computation Conference (GECCO 2024).

The upcoming chapters will thoroughly explore the three research lines. Specifically, the structure of this thesis is as follows: the initial chapter will delve into the field of agent-based models, discussing the chosen technique for researching and analyzing themes. Subsequently, the second chapter will address the first research line, focusing on optimization problems. The third chapter will cover the second research line, i.e. the study of collective behaviors. Lastly, the fourth chapter will extensively examine the third and final research line, centered on the theory of trust.

Each of these chapters devoted to the diverse research lines follows a consistent structure. Beginning with an introduction to the respective theme, the chapters present the models, a thorough exploration of conducted research along with its results and conclusions, and a comprehensive chapter conclusion that ties together the threads of exploration.

A comprehensive summary of my published research is presented in Table 1 . The table outlines the details of each publication, including the year of publication, authors, title, venue, and the research line it falls under. It's important to note that some of the work presented in this thesis is either in submission, submitted, or in-press, as indicated in the 'In' column of the table.

Table 1: List of publications

Year	Authors	Title	In	Research line
2020	Crespi, C., Fargetta, G., Pavone, M., Scollo, R. A., & Scrimali, L.	A game theory approach for crowd evacuation modelling	International Conference on Bioinspired Methods and Their Applications (pp. 228-239). Cham: Springer International Publishing.	Collective Behaviours
2020	Crespi, C., Scollo, R. A., & Pavone, M.	Effects of different dynamics in an ant colony optimization algorithm	7th International Conference on Soft Computing & Machine Intelligence (ISCMI) (pp. 8-11). IEEE.	Optimization Problems
2022	Crespi, C., Fargetta, G., Pavone, M., & Scollo, R. A.	An Agent-Based Model to Investigate Different Behaviours in a Crowd Simulation	International Conference on Bioinspired Optimization Methods and Their Applications (pp. 1-14). Cham: Springer International Publishing.	Collective Behaviours
2022	Crespi, C., Scollo, R. A., Fargetta, G., & Pavone, M.	How a different ant behavior affects on the performance of the whole colony	Metaheuristics International Conference (pp. 187-199). Cham: Springer International Publishing.	Optimization Problems
2022	Crespi, C., Fargetta, G., Pavone, M., & Scollo, R. A.	An Agent-Based Model for Crowd Simulation	Italian Workshop on Artificial Life and Evolutionary Computation (pp. 15-26). Cham: Springer Nature Switzerland.	Collective Behaviours
2023	Crespi, C., Scollo, R. A., Fargetta, G., & Pavone, M.	A sensitivity analysis of parameters in an agent-based model for crowd simulations.	Applied Soft Computing, 146, 110684.	Collective Behaviours
2024	Crespi, C., & Pavone, M.	Does a group's size affect the behavior of a crowd?	Social Simulation Conference (in press). Cham: Springer International Publishing.	Collective Behaviours
2024	Cavallaro, C., Crespi, C., Cutello, V., Pavone, M., & Zito, F.	Group Dynamics in Memory Enhanced Ant Colonies: The Influence of Colony Division on a Maze Navigation Problem.	Algorithms for Network Analysis: Theory and Practice, 17, 2.	Optimization Problems
2024	Crespi, C., Romeo, M., Pavone, M., & Cangelosi, A.	The effects of Trust Dynamics on a multi-agent system.	The Genetic and Evolutionary Computation Conference (submitted).	Trust in multi-agent systems

Chapter 1

Agent-based models

1.1 Theoretical Background

An agent-based model (ABM) is a computational framework designed to simulate the actions and interactions of autonomous agents, whether individual or collective entities. Its purpose is to comprehend the behavior of these agents and the factors influencing their outcomes. A standard ABM generally consists of three key components [17] as depicted in Fig 1.1 which are a set of agents, each distinguished by their attributes and behaviors; a set of relationships defined by specific rules, governing how agents interact with one another and their environment and the environment, that represents the space where interactions occur among agents and between agents and their surroundings.

Widely applied in various scientific fields such as biology, ecology, and social science, ABM employs a "bottom-up" approach. By examining interactions among individual elements within a system, the model seeks to reveal emergent properties resulting from these interactions. These emergent characteristics aren't explicitly predetermined for the entire system; instead, they arise from the independent actions of the agent group. This phenomenon is often described as "the whole is greater than the sum of its parts," where simple agent behaviors, or rules they follow, give rise to complex behaviors and state changes at the system level. In recent years, an increasing adoption of agent-based modeling (ABM) has

Figure 1.1: Schematic representation of the main components of an ABM [79].

been observed. This trend can be attributed to several reasons[53]. First of all, ABM is highly effective in addressing nonlinear individual behavior, agent interactions, and situations where traditional modeling, like differential equations, struggles with complexity and unpredictability. Additionally, it offers a natural description of systems with behavioral entities, representing better reality when modeling, for instance, traffic jams, stock markets, voters, or organizational dynamics. Furthermore, ABM's flexibility allows for easy incorporation of more agents, tuning agent complexity, and adjusting levels of description and aggregation, making it suitable for scenarios where the appropriate complexity level is uncertain [11]. Moreover, the rising complexity of contemporary systems sets limitations of traditional modeling tools in addressing this complexity. ABM, with its ability to model at micro levels and leverage advancements in computational power, emerged as a well-suited solution to navigate the complexities presented by modern systems. Several simulation software options are available for creating Agent-Based Models (ABMs), including AnyLogic, GAMA, Pathfinder, and NetLogo.

For the project implementation it has been used NetLogo, which is also one of the most used [72]. It was developed by Uri Wilensky in 1999 and demonstrates its full potential in creating simulation models for dynamic and complex systems. Indeed, it allows for the replication of numerous characteristics of a complex system, enabling the study of its evolution and real-time visualization within a virtual laboratory. NetLogo's development environment interprets code directly, eliminating the need for compilation. This feature facilitates interaction with the system through buttons and sliders for adjusting control parameters, visualizing graphs related to the simulation, and conducting experiments by varying initial conditions or control parameters.

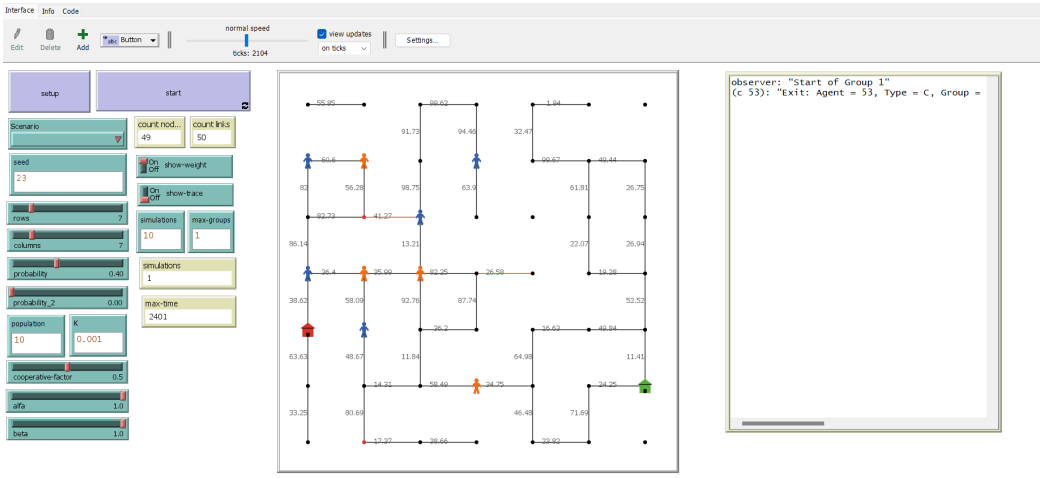


Figure 1.2: In a NetLogo environment, the interface is structured with an array of buttons and sliders on the left, facilitating the adjustment of parameters and the initiation of model runs. Positioned at the center is the world—the environment where agents navigate and interact according to specific relationships. On the right side, the output window displays the printed values of variables of interest. The specific elements, including buttons and sliders and their positions, vary depending on the model being studied, although a setup and a go button are typically constants across all models.

NetLogo was selected as the platform for developing the model of this research project due to its status as both a programming language

and an Integrated Development Environment (IDE). This duality allows for seamless transitions between modeling and simulation perspectives. The familiarity with NetLogo, stemming from previous utilization during the presentation of the master's thesis project, played a significant role in this decision. Consequently, the implementation of the model in each version is intended for NetLogo.

1.2 The model

The model is based on the Ant Colony Optimization (ACO) algorithm, a procedure initially introduced in the 1990s by Marco Dorigo [28] that belongs to the Swarm Intelligence class. Taking inspiration from natural ant colonies' foraging patterns, which can find the shortest path between their nest and food using chemical pheromones, this algorithm has proven to be effective in solving different kinds of combinatorial optimization problems [63] among which scheduling [27], routing [42], coloring [18, 30], robot path planning [4, 12, 100], transportation [43], and feature selection [62].

From the mathematical point of view, the environment in which the ants move is generally represented as a graph $G = (V, E, w)$, where V is the set of vertices, E is the set of edges and $w\colon V \times V \to \mathbb{R}^+$ is the *weighted* function that assigns a positive cost to each edge of the graph. The weight indicates how difficult is to cross a particular edge.

The algorithmic procedure consists of three main rules. The **proportional transition rule** defines the probability $p_{ij}^k(t)$ that ant k on a vertex i selects vertex j as its destination at time t and is expressed as:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in J_i^k} \tau_{il}(t)^\alpha \cdot \eta_{il}^\beta} & \text{if } j \in J_i^k \\ 0 & \text{otherwise,} \end{cases} \tag{1.1}$$

where J_i^k is the potential neighboring vertices of ant k from vertex i, $\tau_{ij}(t)$ represents the pheromone intensity on edge (i, j), and $\eta_{ij}(t)$ denotes the visibility of edge (i, j) at time t (typically defined as the reciprocal of the distance between nodes). The parameters α and β regulate the influence of pheromone intensity versus edge visibility.

The **reinforcement rule** defines the amount of pheromone deposited by ant k on a crossed edge and is expressed as:

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L^{k}(t)} & \text{if } (i,j) \in T^{k}(t) \\ 0 & \text{otherwise} \end{cases} \tag{1.2}$$

where $T^{k}(t)$ represents the path taken by ant k at time t, $L^{k}(t)$ denotes the length of the path, and Q stands as a constant.

The **global updating rule** defines how pheromones evaporate in time and is expressed as:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{1.3}$$

where $\tau_{ij}(t)$ is the pheromone quantity on edge (i,j) from the preceding step, ρ is the pheromone decay parameter and m denotes the number of ants.

In the first research line, the model was employed to investigate the impact of competitive dynamics on optimization algorithms, by using the Ant Colony Optimization algorithm for this purpose. This involved different kinds of analyses of how introducing elements of competition could influence the overall efficiency and outcomes of optimization processes. The goal was to gain a deeper understanding of how competitive dynamics, when integrated into established algorithms like Ant Colony Optimization, could potentially yield positive results in the field of problem-solving and optimization.

In the second research line, the same model was employed to examine collective behaviors by translating ACO principles into the context of crowd dynamics. The objective was to analyze the effects of different behavioral strategies, with a particular emphasis on cooperation and defection, within a crowd context, contributing to the development of a new framework for crowd models. This approach leveraged the inherent adaptability and scalability of ACO-based models, aiming to deepen understanding and provide a comprehensive, adaptable simulation of crowd dynamics.

In the third research line, the model developed in the second research line for collective behaviors was adapted to understand the impacts of

distinct behavioral strategies, placing particular emphasis on trust and skepticism. Specifically, the model was utilized to analyze how agents' trust and skepticism toward a robot influence various path-planning situations in different environmental scenarios

Chapter 2

Optimization Problems

2.1 Introduction

The primary focus of the initial research centers on the fundamental question: Could competition play a role in optimizing outcomes within the context of optimization problems? Before delving into the specifics of the research, it is crucial to establish the necessary context for this exploration.

Optimization problems concerning networks fall under the category of combinatorial optimization and involve a discrete set of feasible solutions. They are typically defined by a quadruple (I, f, m, g), where I represents a set of instances, $f(x)$ denotes the feasible solutions for a specific instance x in I, $m(x, y)$ quantifies a particularly feasible solution y, and g is the function to be optimized. The primary objective in these problems is to find an optimal solution for a given instance x, such that $m(x, y)$ equals the optimal value of g among all feasible solutions in $f(x)$, as expressed in equation 2.1.

$$m(x, y) = g\{m(x, y') \quad | \quad y' \in f(x)\} \tag{2.1}$$

Numerous examples of combinatorial optimization problems fit within this framework, including the shortest path problem, vehicle routing problem, knapsack problem, flow and circulation problems, and supply chain optimization. Over time, various optimization techniques have been de-

veloped to tackle a wide range of practical issues. Some methods, such as exact methods, can provide optimal solutions for smaller problems, while others, like heuristic and metaheuristic techniques, are capable of offering nearly optimal solutions for large-scale problems [40], [76]. In cases where the number of feasible solutions is finite, optimization problems can be addressed using exact methods such as branch and bound or dynamic programming. However, these methods may not always be practical, especially for complex problems like the traveling salesman problem (TSP), which can become computationally infeasible as the problem size increases [87]. The limitations of exact methods become more apparent when dealing with real-world networks that are dynamic and uncertain. In such cases, it's important to quickly adapt to changing conditions and find good solutions within reasonable time frames. This leads to the use of approximate methods, which make random choices and can provide valuable results in situations with a high degree of uncertainty.

A specific class of approximate methods is known as metaheuristics, which are not problem-specific and can be used as black-box tools. Moreover, they are generally non-greedy and can tolerate temporary degradation in solution quality, allowing for deeper exploration of the search space and often resulting in improved solutions (which may coincide with the global optimum).

There is a wide variety of metaheuristic techniques with different classification criteria [10]. One way to classify them is based on the type of search they perform, resulting in *local search* and *global search* algorithms [76]. Local search algorithms may not always find the optimal solution if one exists, while global search algorithms will eventually find the optimal solution given enough time. However, the computational cost of these methods should also be considered. Another classification is based on how the solution is searched, leading to single-solution and population-based algorithms. Single-solution approaches focus on modifying and improving a single candidate solution, whereas population-based approaches maintain and enhance multiple candidate solutions, using population characteristics to guide the search. The most interesting and efficient algorithms, especially for optimization in static and dynamic networks, are undoubtedly the population-based ones (which are also of the "global search" type), and in particular, those belonging to

Swarm Intelligence. SI is defined as the collective behavior of decentralized and self-organized systems in which the group exhibits a form of intelligence greater than that of each individual element. Direct observations of such systems have revealed that a swarm is capable of solving complex problems that a single individual with simple (whether physical or computational) capabilities cannot. As the name suggests, it is composed of numerous elements, some of which may be lost or damaged without affecting the overall group performance. This is because individuals perceive information only at a local level, perform simple actions, have little or no memory, and do not know the global state of the system or its purpose [26].

The Swarm Intelligence (SI) class includes a series of algorithms that can be classified based on their sources of inspiration. There are "insect-based" algorithms (such as ABC and ACO), which are based on the behavior of insects like bees and ants, "bacteria-based" algorithms (like BFOA) inspired by the behavior of bacteria, and even "bird-based" algorithms (like PSO and BA) based on the behavior of birds. What unites these algorithms is their common source of inspiration, which is the natural world and the principles of Darwinian evolution. This is why they are often referred to as "bio-inspired algorithms." Despite differences in names and methods, they share a common dynamic: initializing with a random population of solutions, evaluating them in a loop, selecting some for creating better solutions, and terminating based on a chosen criterion[1].

As previously highlighted, the effectiveness of both real and algorithmic systems in problem-solving stems from the essential cooperation among group elements. This collaborative aspect is vital, generating advantages that individual entities cannot achieve in isolation. Interestingly, studies on the role of competition reveal a paradoxical phenomenon where the presence of competition can enhance cooperation, resulting in improved optimization outcomes.

For instance, in [33], the authors introduced an approach that combines competitive and cooperative mechanisms to address multiobjective optimization problems in dynamic environments. Allowing subpopulations to both compete and cooperate enables the algorithm to adapt to changing optimization requirements over time. Similarly, the Imperialist

Competitive Algorithm (ICA), initially introduced by [90], draws inspiration from socio-political behaviors, involving imperialist competition and simulating empires competing for colonies. This competition leads to the convergence of power, with countries representing individuals forming empires that compete globally. Additionally, the Competitive Swarm Optimizer (CSO) proposed in [35] incorporates a learning mechanism where particles learn from randomly selected competitors, fostering pairwise competitions within randomly divided groups. Winners advance to the next iteration, influencing the loser particles to update their position and velocity based on the winner's behavior. More recently, the Competitive Search Algorithm (CSA) proposed in [91] introduces a competitive mechanism that creates an environment where solutions compete, preventing stagnation in local optima and promoting diverse exploration of the solution space. Overall, CSA enhances search capabilities through competition, facilitating comprehensive exploration of the solution space.

In existing literature, competitive and cooperative mechanisms implementations often involve algorithms with preset parameters, designed and tested for specific instances. This research line, however, introduces a distinctive approach. Using the well-established Ant Colony Optimization Algorithm as a foundation, the study explores the impact of competitive dynamics through an agent-based model featuring two distinct ant colonies—one acting competitively and the other cooperatively.

A key innovation in this research lies in the model's high customizability and adaptability, allowing not only the variation of the percentage of competitive elements within the colony but also the incorporation of different degrees of competition across instances of varying complexity. This flexibility allows an exploration of the algorithm efficiency across a spectrum of different scenarios, adding a new dimension to the evaluation process. Furthermore, the real-time visualization of the simulated dynamics allows an immediate insight into the interactions between cooperative and competitive elements.

This approach sets the stage for two investigations conducted using the same agent-based model with minor variations. Each study addresses different aspects of the research question, employing two ant colonies, one exhibiting competitive behavior, introducing disruptions at specific path points, while the other demonstrated cooperative behavior, rectify-

ing these disruptions. The colonies compete to navigate an unfamiliar environment, contending with disruptions introduced by one and rectified by the other. The first ant model evaluates the consequences of this interplay on colony efficiency using the earned resources metric and assesses algorithm performance through the Success Rate (SR). Subsequently, the second ant model expands upon the initial model, incorporating novel components into ant behavior and conducting additional rounds of data analysis. This extension includes the evaluation of metrics such as the number of ants successfully reaching the exit, exit timing, and path-related costs. Together, these investigations provide a comprehensive exploration of the implications of competitive dynamics in optimization algorithms. In the following sections, both models will be elucidated, the experimental procedures detailed, and the resulting outcomes presented.

2.2 The first ant model

In the first study [25], an existing model [74] was adapted to generate the labyrinth. This model employed an iterative technique known as the Recursive Backtracker, a modified version of depth-first search. Initially designed to create simple and random labyrinths based on specific parameter values, the model was modified by fixing the seed of random numbers. This ensured consistent simulations and allowed the same labyrinth to be regenerated for each run.

Next, a network was established beneath the labyrinth to create more intricate mazes. This altered approach facilitated additional connections between nodes, each having at least two immediate neighbors. This deliberate choice aimed to prevent the removal of dead ends. An entrance and an exit were designated at opposite edges of the labyrinth, with resources positioned at the exit for the ants of the colony to gather. As previously mentioned, the colony consists of:

- Competitive ants: They consistently act competitively, obstructing a random node upon reaching the exit. They only acquire a resource when they identify the shortest path.

- Cooperative ants: They always demonstrate cooperative behavior. When encountering a damaged node along their route, they repair it and collect a resource only upon identifying the shortest path.

Both colonies aim to solve the labyrinth and accumulate the highest number of resources placed at the exit that can be collected if and only if both types of ants reach the exit via the shortest path. The number of ants that act cooperatively is defined by the *cooperation factor* $f \in [0, 1]$. f represents the proportion of cooperative ants with respect to the colony. The remaining ants automatically adopt a competitive approach once the cooperative proportion is established.

The ACO rules were also adjusted to suit the problem being studied. Here, for simplicity, the visibility η_{ij} of a node is not considered ($\beta = 0$). Consequently, the transition probability of an ant k from node i to node j described in the proportional transition rule in Eq. 1.1 depends solely on the pheromone τ_{ij} present on link (i, j) with $\alpha = 1$. About the reinforcement rule in Eq. 1.2, the amount of pheromone left by an ant k after crossing an edge (i, j) was set to a fixed value $\Delta\tau_{ij}^{k}(t) = 1.5$ with the condition that it is placed only by the first ant of each iteration to reach the exit. Subsequently, ants of the same type expire, initiating a new generation. Finally, the global updating rule is incorporated into a process that updates pheromone levels at the end of each successful ant's tour and it is the same as in Eq. 1.3. Additionally, to prevent stagnation of the algorithm and enable continuous exploration of the labyrinth, the pheromone trail on a link cannot drop below a certain nonzero threshold.

2.2.1 Experiments and results

Two distinct analyses were undertaken, focusing on the performance aspects of both the ant colony and the algorithm employed. The first analysis aimed to gauge the colony's efficacy, assessing the quantity of resources earned by the ants. Meanwhile, the second analysis aimed to evaluate the algorithm's performance, centering on its success rate (SR). These evaluations were conducted within the context of two separate labyrinths, each consisting of an identical number of nodes ($|V| = 77$) and links ($|E| = 128$), albeit with varying link distributions. These labyrinths

were designated as Type A, featuring a less uniform link distribution, and Type B, characterized by a more uniform distribution of links. The experiments involved a colony of $N = 100$ ants, where the cooperation factor f ranged from $f = 0.0$ (indicating a colony comprised entirely of competitive ants) to $f = 1.0$ (representing a colony consisting solely of cooperative ants), with incremental steps of 0.05. The initial pheromone level on the links was uniformly set to $\tau_{ij}(0) = 1.0$, while the evaporation rate α was consistently maintained at 0.1. All simulations were capped at a maximum of 100 generations throughout the 1000 conducted simulations.

Performance of the colony

The plots in Fig.2.1a and Fig.2.1b provide insights into the average resources accumulated by the two categories of ants (cooperative and competitive) as well as the overall colony. These averages were computed across 1000 simulation runs. The observed trends in the data align with initial expectations, as higher values of cooperation factor (f) correlate positively with increased resource acquisition. Interestingly, while it might be anticipated that maximal resource gain would occur at $f = 1.0$, the actual data reveals a different pattern. Notably, the pinnacle of earned resources is not reached at the highest cooperation value, but rather at intermediate f values.

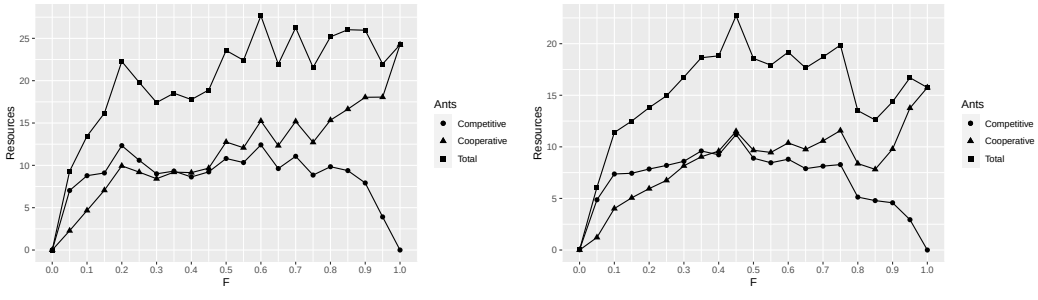
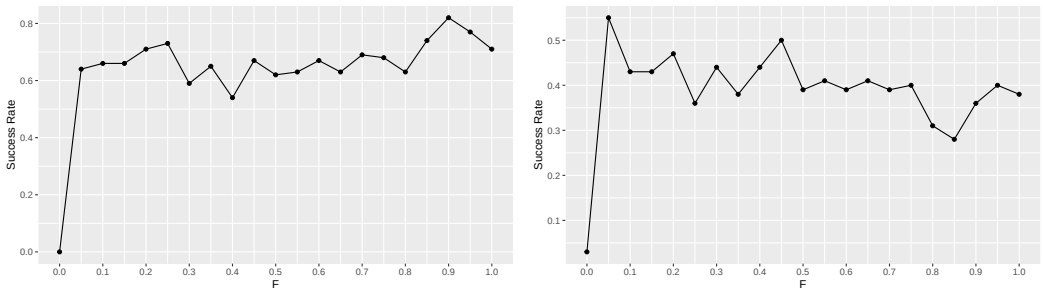


Figure 2.1: Average of the resources earned by the colony on network A (a) and network B (b) calculated over 1000 simulations.



(a) (b)

Figure 2.2: Success rate of the colony on network A (a) and network B (b) calculated over 1000 simulations.

Performance of the algorithm

Turning attention to Fig.2.2a and Fig.2.2b, these plots present the success rate of the colony in a distinct light. Here, the success rate signifies how often the colony manages to identify the shortest path across 1000 simulations. Echoing earlier findings, it becomes clear that the highest success rates are not achieved when $f = 10$.. This seemingly counterintuitive observation occurs in scenarios where all colony ants exhibit cooperative behavior.

2.2.2 Conclusions

The analysis of the data reveals a noteworthy insight – the introduction of a small proportion of competitive agents brings advantages to the entire colony. Addressing the research question of whether competition can contribute to the optimization of outcomes in the context of optimization problems, the results demonstrate that relying solely on cooperative ants does not lead to optimal outcomes. Instead, the presence of certain ants exhibiting competitive behavior proves crucial for achieving superior results. This seemingly counterintuitive finding can be attributed to the behavior of competitive ants reaching the exit, obstructing a path node, and thereby prompting the rest of the colony to adapt and explore alternative paths.

2.3 The second ant model

In the second study [23], a new procedure was employed to create more intricate and varied environments. The environment in which the ants navigate is now a weighted network represented as $G = (V, E, w)$ where V is the set of vertices, E is the set of edges, and $w\colon V \times V \to \mathbb{R}^+$ functions as the weighted operation, assigning a positive cost to each edge within the graph. The weight indicates the level of challenge associated with traversing a specific edge. A starting point is randomly selected from one side of the graph, such as the left side, while an exit point is also randomly chosen on the opposite side, like the right side. It's important to note that every link is traversable in both directions.

As previously mentioned, novel components were introduced to modify the behaviors of the ants in relation to the first study. In this context, the ant colony is comprised of two distinct types of ants:

- High Performing Ants (HPAs): These ants consistently demonstrate high levels of performance. When encountering a damaged edge and/or vertex, they possess the ability to repair it, and this repair is influenced by a probability range of $0 \leq \rho_{e,v} \leq 1$. Furthermore, HPAs contribute two distinct forms of information concerning their path: They deposit the conventional pheromone $\tau_{ij}(t)$ upon successfully traversing an edge (i, j) and a more intricate form of information referred to as $\eta_{ij}(t) = 1/w_{ij}(t)$. In this context, $w_{ij}(t)$ represents the weight of the edge (i, j) at a specific time t, and $\eta_{ij}(t)$ serves to communicate to the rest of the ant colony the level of difficulty associated with that particular path.

- Low Performing Ants (LPAs): These ants consistently exhibit suboptimal performance. Their functionality is compromised, leading to the potential destruction of certain edges and/or vertex within the network. This destruction occurs with a specific probability range of $0 \leq \rho_{e,v} \leq 1$. Notably, LPAs leave the classical pheromone $\tau_{ij}(t)$ after crossing an edge but do not leave any piece of information $\eta_{ij}(t)$ behind after traversing an edge (i, j) along their designated path.

It is noteworthy that the act of destroying an edge or node renders it unfeasible for traversal. Conversely, repairing an edge or node restores its practicability. Both actions contribute to the network's dynamism. In this context, the second study can be viewed as an extension of the preceding general shortest path problem of the first study. This perspective led to the decision to exclude the shortest path as an evaluative metric due to the network's dynamic nature. Consequently, in this scenario, both categories of ants are tasked with locating the exit point of the network from a designated entrance. Their objective, however, is to optimize the number of ants reaching the exit while minimizing path cost and resolution time. The quantity of High-Performing Ants (HPAs) within the colony is determined by the parameter denoted as the performing factor (p_f), which lies in the interval $[0, 1]$. Consequently, once is fixed, the remaining ants (namely, $1 - p_f$ fraction) constitute Low-Performing Ants (LPAs). It should be noted that the situation where $p_f = 1.0$ corresponds to the complete presence of HPAs, resulting in the classical version of ACO. Let be $A_i = \{j \in V : (i, j) \in E\}$ the set of vertices adjacent to vertex i and $\pi^k(t) = (\pi_1, \pi_2, \ldots, \pi_t)$ a non-empty sequence of vertices, with repetitions, visited by an agent k at the time-step t, where $(\pi_i, \pi_{i+1}) \in E$ for $i = 1, \ldots, t-1$. Due to the influence of HPAs, which possess the capability to restore damaged nodes and/or links, the path $\pi^k(t)$ becomes more intricate than a straightforward path. This complexity arises from the possibility of an ant revisiting a vertex through a back-tracking operation.

Also in this second study, the ACO rules were tailored to suit the current problem being examined. The probability $p_{ij}^k(t)$ of an ant k positioned on a vertex i selecting one of its neighboring vertices j as its destination at time t is determined using the proportional transition rule defined in Eq. 1.1. Here, $J_i^k = A_i \setminus \pi_t^k$ represents all feasible movements of ant k from vertex i while the visibility $\eta_{ij}(t)$ is called desirability and establishes how much an edge (i, j) is promising. In particular and it is defined as $\eta_{ji}(t)) = \frac{1}{w_{ij}(t)}$. This information is released by each ant as the trace, however, it does not depend on the ant itself, but only on the edge (i, j). Each ant leaves an amount of pheromone K after crossing an edge (i, j) according to the reinforcement rule as defined in Eq. 1.2. For contextual clarity within the scenario, the term pheromone will be sub-

stituted with the term *trace*. Pheromones evaporate in time every T ticks [1] according to the global updating rule as defined in 1.3.

The goal of the ants is to explore the network and find the most economical path from the starting point to the endpoint in the least amount of time. They achieve this by using the amount of traces on paths and the exchanged information about desirability. Simultaneously, the aim is to maximize the number of ants reaching the endpoint. This entails the optimization of three distinct objective functions: minimizing the path cost function and the exit time function, and maximizing the exit function, indicating the number of ants that successfully reach the endpoint. To streamline the optimization process, the path cost function and the time cost function have been combined into the following unified objective function:

$$\min \sum_{i=1}^{t-1} w(\pi_i^k, \pi_{i+1}^k) + |\pi^k|. \tag{2.2}$$

Finally, the exit function is defined as:

$$\max \sum_{g \in G} \sum_{k \in N} k_g. \tag{2.3}$$

It represents the maximization of the number of ants that must reach the exit, where G is the total number of groups, g is the index of the group to which the ant k belongs, k_g is the ant k that belongs to g group and N is the set of ants.

2.3.1 Experiments and results

The simulations have been conducted using two distinct types of scenarios, each corresponding to networks of increasing complexity. In each scenario, two model parameters have been adjusted. Specifically, the quantity K of pheromone deposited by the ants on the links, and the parameter β,' which gauges the importance of information $\eta_{ij}(t)$ in relation to the actual pheromone quantity. The two scenarios are:

- *Scenario B1*: a 15x15 network with $|V| = 225$ nodes and $|E| = 348$ links.

[1] Each tick corresponds to an ant displacement and movement.

- *Scenario B2*: a 15x15 network with $|V| = 225$ nodes but $|E| = 495$ links.

The general experimental setup is as follows. For each scenario, a total of $N = 1000$ ants have been utilized, distributed across $G = 10$ groups. This implies that each group comprises $N_g = 100$ ants, commencing their journey from the initial point at regular intervals calculated by multiplying row and column values, resulting in $T_l = 225$ ticks. As previously mentioned, the ant colony consists of two distinct types: high-performing ants (HPAs) and low-performing ants (LPAs). The proportion of HPAs and LPAs is governed by a performance factor p_f," determining the ratio of the former to the latter. This factor ranges from $p_f = 0.0$ (representing a colony exclusively composed of LPAs) to $p_f = 1.0$ (constituting a colony solely comprising HPAs, analogous to the classical ACO version), with increments of $p_f = 0.10$. The goal for the ants is to find the exit within a certain time frame T_{max} calculated as $2 \times G \times T_l$, where G is the group count, T_l is the launch interval, and 2 adjusts the value. Initially, the links have a trace intensity of 1.0, which diminishes over time every $T_d = 50$ by 10% ($\rho = 0.10$). In scenarios labeled as "High Trace", both α and β parameters are set to 1.0. Additionally, each ant releases an amount of pheromone $K = 0.1$. For scenarios classified as "Low Trace," α remains 1.0, while β is adjusted to 0.5. In this case, each ant leaves less trace on its path, with an amount of $K = 0.001$. The parameter β influences the balance between the impact of information η_{ij} and trace τ_{ij}. By decreasing both β and K, the colony's behavior is expected to rely more on the acquired path information and less on the trace left behind. The probabilities for edge destruction-repair ($\rho_e = 0.02$) and vertex destruction-repair ($\rho_v = 0.02$) are consistent across both configurations. A series of 10 independent simulations have been conducted, ranging from an initial performance factor value of $p_f = 0.0$ up to $f = 1.0$, in increments of 0.1. Finally, two analysis types have been carried out:

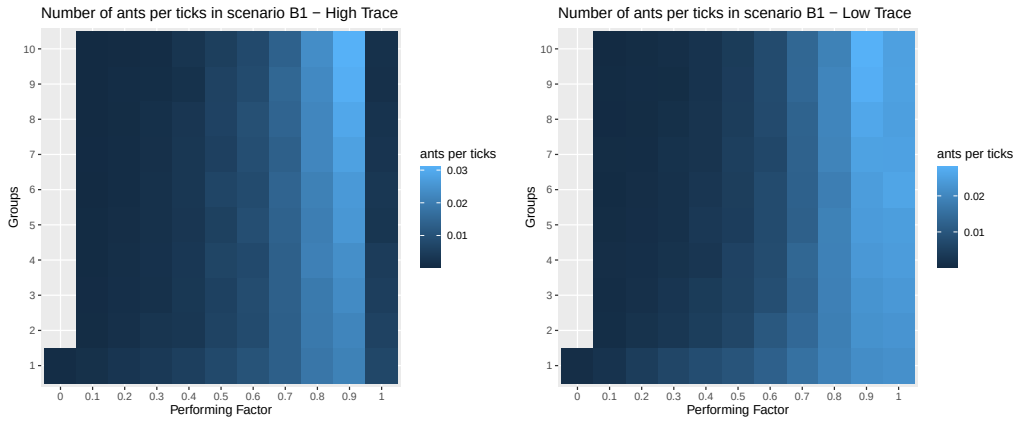
- **Group Analysis**: This examines the number of ants that successfully reach the exit. It considers the performing factor value and the group count.
- **Overall Analysis**: This assesses the following aspects: The cost of

the path determined by the ant colony; the time taken by the ants to find the path and the number of ants that reach the exit.

In the forthcoming results, the term High Trace corresponds to a K value of 0.1 and a β value of 1.0". On the other hand, the label Low Trace corresponds to a K value of 0.001 and a β value of 0.5.

Group Analysis

In this initial analysis, the number of ants reaching the exit is evaluated based on the performing factor and the number of groups. A heatmap illustrates the results, with lighter shades of blue indicating higher ant numbers, and darker shades indicating lower numbers. The absence of color means no ants reached the exit for specific performing factor or group values. The results for scenario B1 are presented in Fig. 2.3, where Fig. 2.3a shows ants per tick with a high-level trace, and Fig. 2.3b depicts the same with a low-level trace. The comparison between Fig.2.3a and Fig.2.3b reveals significant differences in optimal outcomes for ant colonies under varying conditions. Specifically, with a high-level trace, the most favorable ant colony performance is achieved by the later groups when the performing factor is approximately $p_f = 0.9$. Conversely, in the presence of a low-level trace, optimal results are seen with the last groups not only at $p_f = 0.9$ but also at $p_f = 1.0$. This suggests that the influence LPAs is more pronounced when the trace level is high. This effect is especially noticeable when comparing ant exit rates at performing factors around $f = 0.9$ or slightly lower, regardless of group differences. These findings are attributed to the ants' tendency to be misled by the strong trace in high-level scenarios, prompting them to follow incorrect paths. Furthermore, analysis of Fig. 2.3 reveals that a high-level trace leads to a greater number of ants reaching the exit compared to a low-level trace. This underscores the critical role of LPAs in optimizing ant colony performance under high-level trace conditions, as indicated by the poorer colony outcomes at $p_f = 1.0$. Fig.2.4 presents an analysis analogous to that conducted for scenario B2, showcasing a pattern reminiscent of the previous heatmaps in scenario B1, albeit with some distinctions. In the instance of a high-level trace, illustrated in Fig.2.4a, the overall count of



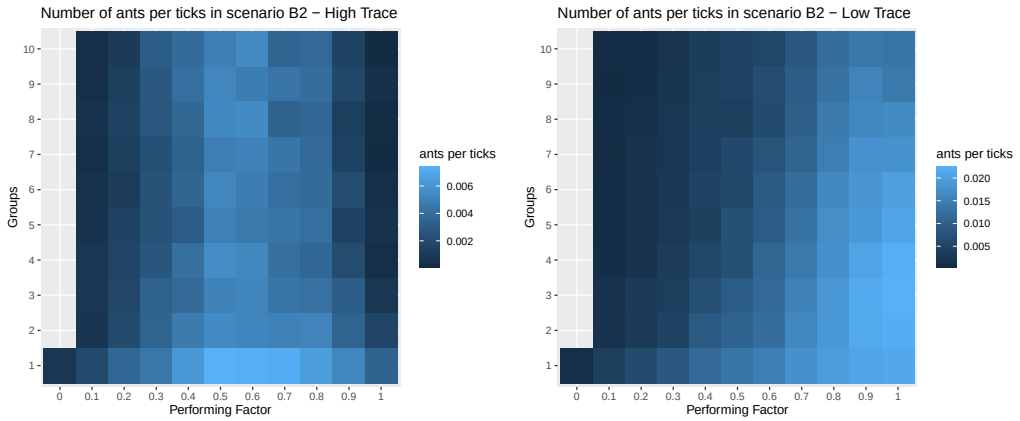
(a) (b)

Figure 2.3: The heatmap illustrates the number of ants reaching the exit per tick in scenario B1. The colony's performance varies based on the trace amount released by the ants. In (a), the best results occur for the final group ($g = 10$) with a performing factor of $p_f = 0.9$ when a high-level trace is present. A similar trend is seen in (b) for a low-level trace, where good performance persists up to a performing factor of $p_f = 1.0$

ants reaching the exit tends to be lower than that observed under the same configuration in scenario B1. Notably, the most favorable outcomes occur when the initial colony groups exhibit a performing factor around $p_f = 0.5$, which includes a portion LPAs. Conversely, when confronted with a low-level trace, as seen in Fig. 2.4b, the optimal result is achieved by the first group when the colony is predominantly composed of HPAs ($p_f = 1.0$). This observation reinforces the idea wherein a high trace level leads to colony confusion, and simultaneously, a minor presence of LPAs prompts behavioral shifts within the rest of the group.

Overall Analysis

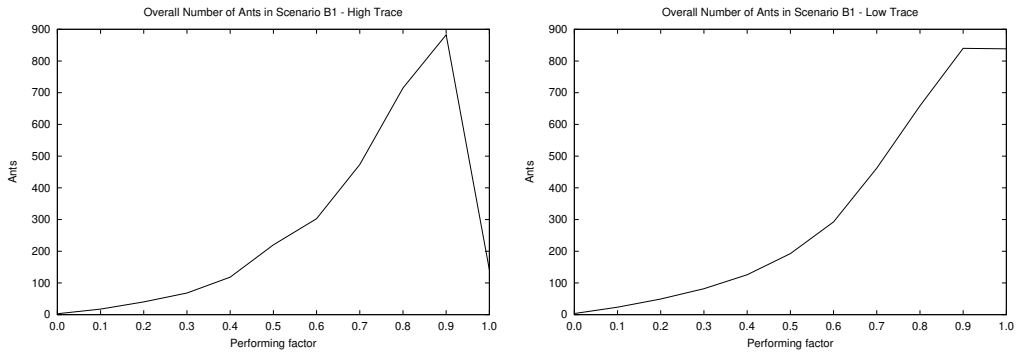
This analysis focuses solely on how colony performance changes with the performing factor, without considering the number of ant groups. It examines different colony performances under high-level and low-level traces. The study looks at the number of ants reaching the exit, path cost,



(a) (b)

Figure 2.4: The heatmap depicts ant behavior in scenario B2, measuring the number of ants reaching the exit per time unit. For a high-level trace (Fig.2.4a), optimal colony performance is achieved by the first groups when the performing factor is around $p_f = 0.5$. In contrast, a low-level trace (Fig.2.4b) leads to better results with higher-performing factor values, exceeding $p_f > 0.7$, and extending the favorable outcomes to more groups beyond the initial ones.

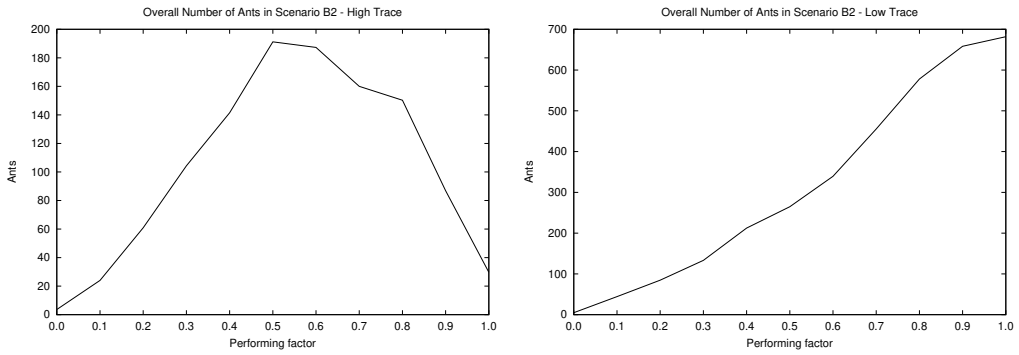
and resolution time, aiming to maximize ant count while minimizing path cost and resolution time. In scenario B1, Fig.2.5 displays the ants reaching the exit. Fig.2.5a shows that LPAs are more effective when there's more trace, resulting in better outcomes when only a small percentage of LPAs exist (best at $p_f = 0.9$). Yet, under a low-level trace in Fig. 2.5b, LPAs don't significantly improve colony performance; ant count at exit remains similar for $p_f = 0.9$ and $p_f = 1.0$. The results for scenario B2 are presented in Fig.2.6. When there is a high-level trace, Fig.2.6a displays the number of ants reaching the exit. The maximum ant count occurs at a performing factor of $p_f = 0.5$. Conversely, Fig. 2.6b illustrates this count for a low-level trace, with optimal colony performance at $p_f = 1.0$. Just like in the previous case, the presence of LPAs appears more impactful when an excess trace is released on the path. This suggests improved colony performance with mixed ant types rather than just HPAs. Interestingly, scenario B2 yields lower average ant numbers compared to scenario B1,



(a) (b)

Figure 2.5: Overall number of ants that have reached the exit in scenario B1. In (a) the values obtained for a high-level trace; in (b) the ones obtained for a low-level trace. The presence of LPAs is much more important and useful when there is a high-level trace, leading the colony to better performances. The best values are obtained for $p_f = 0.9$ when there is a high-level trace and for $p_f = 1.0$ when there is a low-level trace.

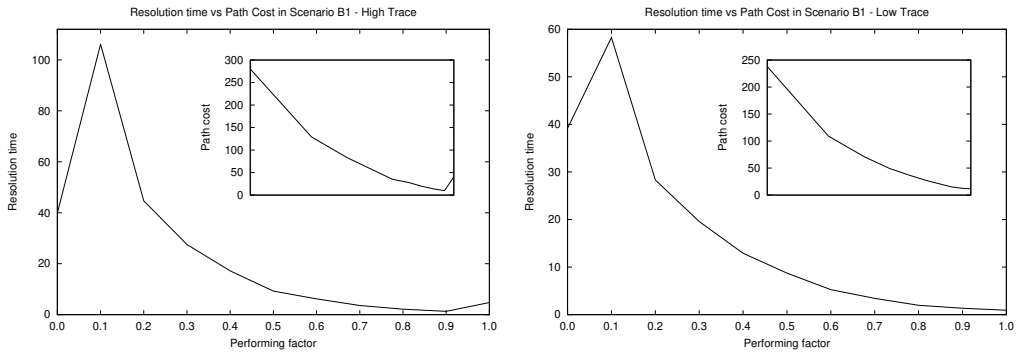
possibly due to network complexity: higher complexity corresponds to poorer colony performance. The resolution time and path cost, both to be minimized, are shown together in a single plot. The main graph illustrates the resolution time, while the inset graph depicts the path cost. This kind of presentation is applied to both scenario B1 (Fig.2.7) and scenario B2 (Fig.2.8). Specifically, Fig.2.7a displays resolution time and path cost with respect to the performing factor in scenario B1 with a high-level trace. Similarly, Fig.2.7b shows these quantities under the same scenario with a low-level trace. Optimal values are the lowest ones, reflecting superior colony performance. By comparing these outcomes with the ant count results in Fig.2.5a, it becomes apparent that in scenario B1 with a high-level trace, the colony performs best with a small fraction of LPAs. This is evident because the maximum ant count reaching the exit, the minimum resolution time, and the minimum path cost are achieved when the performing factor is $p_f = 0.9$. Similarly, when examining Fig.2.7b alongside Fig. 2.5b, it becomes clear that in the presence of a low-level trace, LPAs neither positively impact the number of ants reaching the exit



(a) (b)

Figure 2.6: Overall number of ants that have reached the exit in scenario B1. In (a) the values obtained for a high-level trace; in (b) the ones obtained for a low-level trace. As in Fig. 2.5, the presence of LPAs is much more helpful when there is a high-level trace. The best values are obtained for $p_f = 0.5$ when there is a high-level trace and for $p_f = 1.0$ when there is a low-level trace.

nor the resolution time and path cost determined by the colony. In this case, optimal results are attained when the colony comprises only HPAs, further supporting the hypothesis that LPAs are valuable in regulating actions during trace excess. The study found consistent results for scenario B2, as depicted in Fig. reffig:B2-resolution-time-path-cost. In this scenario, the impact of LPAs on the colony's performance is evident, particularly when high-level traces are present. When the performing factor is $p_f = 0.9$, the colony's path cost is significantly improved compared to $p_f = 1.0$, as shown in the inset plot. This performance difference is also reflected in resolution time and the number of ants, displayed in Fig. reffig:B2-ants-high-trace. However, for low-level traces, LPAs do not enhance colony performance; instead, colonies composed solely of HPAs perform better. Fig. reffig:B2-resolution-time-path-cost confirms that the optimal values for resolution time and path cost occur at $p_f = 1.0$.

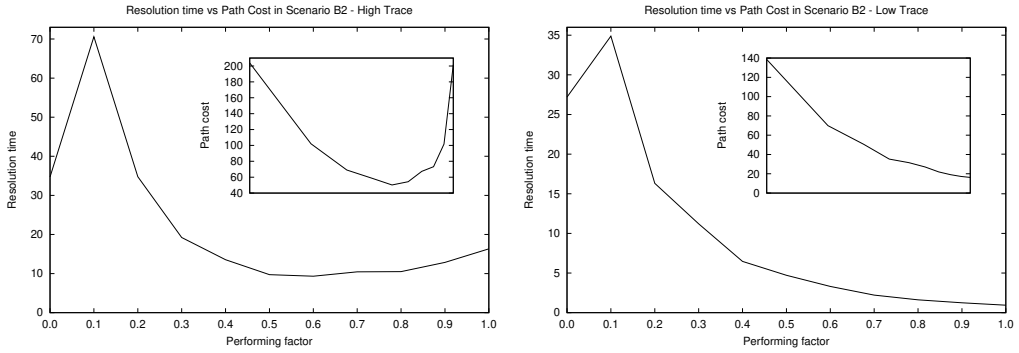


(a) (b)

Figure 2.7: Overall resolution time (principal plot) and path cost (inset plot) of the colony for scenario B1. In (a) the values obtained for a high-level trace; in (b) the ones obtained for a low level trace. As in Fig. 2.5, the presence of LPAs is much more helpful when there is a high-level trace. The best values are obtained for $p_f = 0.9$ when there is a high-level trace and for $p_f = 1.0$ when there is a low-level trace.

2.3.2 Conclusions

In this second study within the optimization research line, the findings reveal that LPAs provide notable benefits (particularly in scenarios with high levels of trace), disrupting typical path choices and prompting ants to explore alternative routes, fostering information sharing among colony groups. This effect holds even when considering overall colony objectives. LPAs play a crucial role when there is an abundance of traces shared by ants in the environment. Too much trace can hinder the colony's effectiveness by causing ants to repeatedly choose the same path due to their actions being influenced by excessive trace. LPAs counteract this by compelling the colony to alter its behavior, encouraging the exploration of more productive paths. This observation resonates with the conclusions drawn from the first study, where introducing a small proportion of competitive agents demonstrated advantages in optimizing outcomes. The collective impact of LPAs altering the colony's behavior to explore more productive paths aligns with the contribution of competition highlighted in the first work.



(a) (b)

Figure 2.8: Overall resolution time (principal plot) and path cost (inset plot) of the colony for scenario B1. In (a) the values obtained for a high-level trace; in (b) the ones obtained for a low-level trace. As in Fig. 2.6, the presence of LPAs is much more helpful when there is a high-level trace. When there is a high-level trace, the best value of the resolution time is for $p_f = 0.6$ and the one for the path cost is for $p_f = 0.5$. When there is a low-level trace the same bests are obtained for $p_f = 1.0$.

2.4 Conclusions

In summary, the primary objective of the first research line was to investigate the influence of competitive dynamics on optimization algorithms, specifically focusing on the Ant Colony Optimization algorithm. To comprehensively explore this impact, two distinct ant models were developed, employing an agent-based model approach. The overarching question guiding this investigation was:

1. *Could competition contribute to the optimization of outcomes in the context of optimization problems?*

   Yes, competition, when strategically integrated, can enhance algorithmic efficiency.

   In the initial ant model, the introduction of competitive ants resulted in positive effects, notably enhancing the algorithm's success rate. Their strategic blocking actions at crucial nodes prompted the

colony to adapt, explore alternative paths, and ultimately improve overall performance.

Likewise, in the second ant model, the presence of LPAs (competitive ants) positively influenced efficiency, especially in environments rich with trace information. However, this also highlighted the intricate interplay between cooperative and competitive dynamics, particularly emphasized in scenarios with high trace information levels. This insight emphasizes the importance of competition, especially in environments where the algorithm can strategically adapt and optimize its performance based on environmental characteristics.

Challenging the traditional approach, the proposed models revealed that, on the one hand, competition, when strategically integrated, can enhance algorithmic efficiency, and on the other hand, the agent-based approach can be a powerful tool that enables the easy exploration of algorithm efficiency across a variety of scenarios.

Chapter 3

Collective Behaviors

3.1 Introduction

The second research line directed its attention toward dissecting the complexities inherent in collective behaviors within social complex systems. This research was driven by a fundamental question: How do cooperation and competition interact in the dynamics of social complex systems? To explore the complexities of social interactions, the study turned to crowd simulations, acknowledging them as a valuable tool. However, before delving into the detailed investigations and contributions, it is essential to lay down the foundational context for a more profound exploration of this second research line.

Crowd simulations have gained increasing attention in recent years due to their numerous applications, such as emergency management [51, 71], sociology [60, 83], computer games [99], and path planning [85] among many others. This has led to a wide range of models and techniques being developed in the field, with significant progress made possible by advancements in computer hardware and software. Despite the increasing literature available, creating crowd models is still challenging for several reasons. Firstly, there is a lack of data available for modeling crowd behavior, which can limit the accuracy of models [61, 72]. Secondly, crowd behavior is a complex and dynamic phenomenon, that may be influenced by a wide range of factors such as age, environment, and

personality [80, 48]. These factors can significantly shape individual and collective behavior because interactions and dynamics among individuals become more complex due to the simultaneous presence and interaction of such factors. For example, in [78] the authors propose an agent-based model of emergency evacuation that takes into account panic behaviors. In [3] are considered the physical characteristics of pedestrians while in [49] a specific emergency is taken into account. In [101], the authors employ a game-theoretical model to elucidate the interactions between cooperative and competitive agents within a crowd. It follows that there exist various crowd behavior models, and each one brings its own set of assumptions, methodologies, and theoretical perspectives, leading to different interpretations and insights into crowd behavior. Due to the intricate nature of crowd behavior, no single model can fully capture all its aspects.

Reviews and surveys [72, 93] have attempted to categorize these models into three main types: microscopic, macroscopic, and mesoscopic. Microscopic models [81, 102] consider individuals as distinct entities with unique traits whose interactions may produce unexpected collective behaviors, while macroscopic models [98, 54] view individuals as a continuous flow generally governed by adequate physical dynamics. Although microscopic models are adept at studying collective behavior, they are computationally expensive and not well-suited for modeling large-scale scenarios. This is primarily because of their inherent computational complexity and memory requirements. Conversely, macroscopic models struggle to capture individual interactions but can simulate thousands of individuals. Mesoscopic models aim to combine the strengths of both macro and micro techniques [95].

Building upon existing literature on crowd simulation, the primary objective was to develop a model capable of analyzing the effects of various behavioral strategies within a crowd context, with a particular focus on cooperation and defection. In pursuit of this objective, a secondary question emerged: *can principles from the Ant Colony Optimization (ACO) algorithm, previously utilized in the first research line, be effectively applied to simulate social dynamics?* This inquiry stems from the demonstrated utility of swarm intelligence algorithms, not only for optimization purposes but also in modeling crowd dynamics [39, 103, 86]. Leveraging insights from

the ACO algorithm, the initial phase of establishing this second research line involved identifying an appropriate abstraction process to map its rules onto conceptual frameworks rooted in social dynamics and agent-based simulations. The ACO principles offer several advantages for this endeavor: they facilitate the emergence of complex behaviors from simple interactions, adaptively model dynamic decision-making processes through pheromone-based communication and reinforcement learning, and ensure scalability to accurately represent large-scale crowd simulations. This mapping process represents a novel contribution to this research line, laying the foundation for subsequent investigations into crowd behaviors.

By examining the core principles of the ACO algorithm, particularly its proportional transition rule, reinforcement rule, and global updating rule, a parallel framework for modeling decision-making and environmental interaction within crowd simulations was established. In detail, given that the proportional transition rule 1.1 of ACO governs how ants make path choices based on factors such as the concentration of pheromones $\tau_{ij}(t)$ along a route and the visibility η_{ij} of a node, a parallel approach was adopted to model how agents or evacuees make decisions regarding their routes in an unfamiliar environment. The main difference lay in the interpretation of pheromones as traces inadvertently left behind by the agents during their movements, and visibility as a measure of desirability associated with a path. This desirability is intentionally communicated to neighboring paths to assist others in identifying optimal routes. Similarly, just as the reinforcement rule 1.2 governs the quantity of pheromone deposited by an ant upon traversing an edge, the same rule is employed in this context to depict the interaction between agents and their environment. In this scenario, agents leave inadvertent traces as they navigate a path. These traces reflect the frequency with which an edge along the path has been traversed, but they lack a physical interpretation. Instead, they represent information about the prior actions of agents and the state of the environment. In simpler terms, think of it as agents leaving behind footprints as they cross a path. These footprints tell where agents have been and how often they've been there, giving insights into their actions and the conditions of their surroundings. Lastly, within the framework of the ACO algorithm, the global updating rule 1.3

governs the gradual evaporation of pheromones on the edges over time. In the model, this rule simulates the impact of time on the environment. Specifically, it controls the gradual reduction of the traces left by agents as time elapses, thereby influencing how the information about the environment undergoes alteration. To elaborate further, it dictates how the marks left by agents diminish over time, mirroring the way knowledge about the environment evolves as it interacts with the temporal dimension.

To summarize, the mapping process consists of the following phases, also outlined for clarity in Table 3.1: pheromones are interpreted as traces, and node visibility as path desirability. The proportional transition rule mirrors how agents make decisions, while the reinforcement rule depicts their interaction with the environment through traces. The global updating rule simulates how time influences traces and the evolution of environmental knowledge.

Table 3.1: Comparison of ACO Rules and Crowd Model

ACO Rule	**Crowd Model**	**Explanation**
Proportional Transition Rule (1.1)	Decision-making in an unfamiliar environment	Agents make decisions about paths based on trace intensity and path desirability
Reinforcement Rule (1.2)	Interaction between agents and environment	Agents leave traces (footprints) reflecting path traversal frequency
Global Updating Rule (1.3)	Gradual degradation of traces	Simulates the impact of time on the environment, controlling the reduction of agent traces over time

After establishing the mapping, the second phase of this research line involved the development of the model, which will henceforth be referred to as the crowd model. This phase involved the creation of three versions

of the crowd model, each with similarities but also marked distinctions. These versions are presented in the order of their evolutionary progression. Despite these variations, the core model revolves around two distinct types of agents navigating a virtual environment from a predetermined starting point to an exit. These agents demonstrate contrasting behaviors: collaborative agents proceed cautiously toward the exit while assisting others, whereas defectors pursue individual and often reckless paths.

In detail, the first study delved into a generic evacuation process from a game theory perspective, analyzing the path choices of agents using a specific profit function. This profit function, within the context of game theory, attributes value to the emotional or economic gain associated with an agent's strategy adoption. In the second, third, and fourth studies, the research evolved to encompass a broader array of crowd scenarios. Evaluation criteria were expanded to include metrics such as the number of agents successfully exiting the environment, path costs, and exit times assessments. Analyses evolved to consider not only the performance of the two agent types but also the various groups they were divided into, the collective behavior of the entire crowd, and detailed sensitivity analysis of the parameters utilized. Finally, in the fifth study, the agents were endowed with a form of memory, enabling them to incorporate past experiences into their decision-making processes. This addition introduced a new layer of complexity to the evaluations but the same set of evaluation metrics was applied to conduct analyses, ensuring uniformity and comparability across all studies.

Overall, in all crowd models, the decision-making processes of the agents, their interactions with the environment, the temporal aspect, and its influence on the environment were all modeled using the aforementioned mapping.

3.2 The first crowd model

The initial model, published in [21], considered a hypothetical high-risk scenario where an unspecified group of agents is assigned the challenging task of navigating an unfamiliar environment, represented as a graph

denoted as $G = (V, L)$ with V representing the set of vertices and L the set of links. Their ultimate objective is to reach a secure location for their survival. This secure location, referred to as the "shelter," comes with a capacity that diminishes as each agent arrives. If the shelter reaches its capacity, meaning it is fully occupied, the exit relocates to another edge node within the graph. This dynamic scenario necessitates not only the agents' ability to find the exit through the shortest path but also their capacity to adapt periodically to new objectives. In adopting a game theory approach, this high-risk scenario is framed as a game involving N players (evacuees), with N being greater than or equal to 2. In this context, players assumed to be rational and intelligent, seek to maximize their profit function which represents the benefits of adopting specific strategies. Within the model, agents can choose between two distinct strategies:

- The Non-Cooperative Strategy: Agents have the option to disrupt a randomly selected node along their path, rendering it impassable.
- The Cooperative Strategy: Agents can choose to repair a damaged node located near their path.

In simpler terms, the non-cooperative strategy reflects agents acting urgently, as if they are in a state of panic, potentially posing risks to themselves and the environment. Conversely, the cooperative strategy encourages agents to act more cautiously, considering their well-being and that of the environment, and taking corrective actions when necessary. Safety for both types of agents is guaranteed only if they successfully reach the exit.

One of the key insights gained during the development of this initial model was the importance of finding a balance between realism and simplicity in crowd modeling, a realization drawn from extensive literature research. While real-world crowd behavior is influenced by diverse factors such as intentions, knowledge, and motivations, attempting to incorporate every aspect into a model can lead to computational complexity and challenges. Hence, the collaborative and non-collaborative strategies were implemented straightforwardly, serving as abstractions to represent a spectrum of behaviors realistic within a crowd.

Formally, each player's payoff function is defined, mapping strategy profiles to numerical values. In this case, two payoff functions are defined, one for each strategy. The payoff function for an agent who chooses the cooperative strategy (a_k^C) is represented as follows:

$$u_k(a_k^C, a_{-k}) = \frac{f \cdot \sum_{i,j} \tau_{ij}}{n}, \quad 0 < f \leq 1. \tag{3.1}$$

The payoff function for an agent who chooses the non-cooperative strategy (a_k^{NC}) is defined as:

$$u_k(a_k^{NC}, a_{-k}) = \frac{(1-f) \cdot \sum_{i,j} \tau_{ij}}{n}, \quad 0 \leq f < 1. \tag{3.2}$$

Here, τ_{ij} represents the trace intensity on an edge (i, j), and $\sum_{i,j} \tau_{ij}$ is the sum of the trace on the links of the agent's path. Moreover, f denotes the percentage of cooperative players and is a user-defined parameter. Once it is defined, the other $1 - f$ agents will adopt a non-cooperative strategy. Finally, n is the number of evacuees in a group, and a_k denotes a generic strategy that agent k can choose from (C) or (NC), with a_{-k} representing the strategies of all agents except agent k. The payoff functions for all cooperative (C) and non-cooperative players (NC) are obtained by aggregating for all k as follows:

$$u^C = f \cdot \sum_{i,j} \tau_{ij}, \quad 0 < f \leq 1; \qquad u^{NC} = (1-f) \cdot \sum_{i,j} \tau_{ij}, \quad 0 \leq f < 1.$$

Ultimately, the profit function of the game is the sum of the payoff of all cooperative ants plus the payoff of all non-cooperative ants, i.e., $U = u^C + u^{NC}$.

3.2.1 Experiments and results

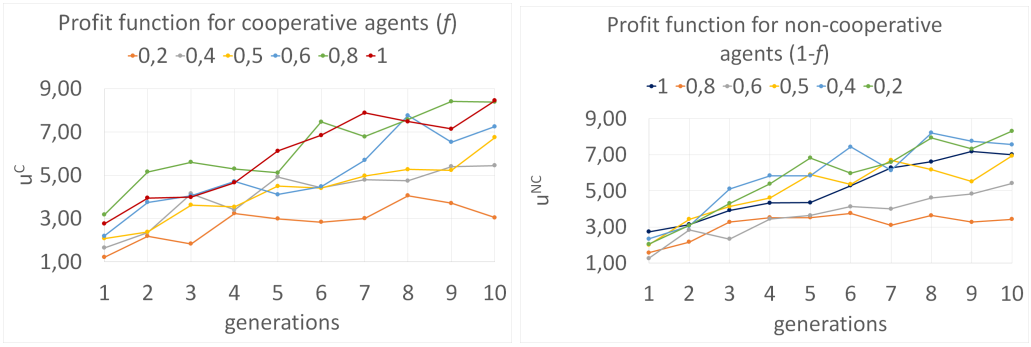
A population of $n = 100$ agents was utilized, distributed across $g = 10$ groups. Consequently, each group comprises 10 agents, and the exploration begins once the first agent of the group arrives at the shelter. It's important to note that priority is exclusively granted to the first agent from each group upon their arrival at the shelter, and the other agents

within the same group do not have the option to share the same shelter. The prioritization of the first agent from each group upon reaching the shelter serves as an abstraction to model a strategic approach aimed at optimizing and maintaining order in scenarios characterized by limited resources. In situations such as emergencies or when shelter capacity is restricted, this approach ensures that all groups are provided with an equitable opportunity to access critical resources, including space, supplies, and assistance. The environment in which the agents move is a graph with ($|V| = 77$) nodes and ($|E| = 128$) links. For simplicity in this first study, the agents choose their next path taking into account only the amount of trace τ_{ij} present on the edges and not the visibility η_{ij}. Formally, this means that the β parameter in the proportional transition rule 1.1 is set to zero while $\alpha = 1.0$. The amount of trace left by the agents after they cross an edge is expressed by the reinforcement rule 1.2 and in this case, it has a fixed value $\Delta\tau_{ij} = 1.5$ while $\tau_{ij}(0)$ the initial amount of trace is set to $\tau_{ij}(0) = 1.0$. Finally, the evaporation rate in the 1.3 is $\rho = 0.1$. With these defined parameters, a series of 10 independent simulations were conducted, encompassing a range of values for f. This range goes from 0.0 to 1.0, with increments of 0.20 introduced at regular intervals. The primary objective was to analyze how the profit functions of the two types of agents evolved concerning the groups that explored the environment. To do this, two kinds of analysis were carried out:

- **Generation type analysis**: To examine how the average profit function of cooperative and non-cooperative agents changed over generations.
- **Overall type Analysis:** to assess how the average profit functions of both cooperative and non-cooperative agents changed with respect to the cooperative factor f.

Generation type analysis

In this analysis, the primary focus was on understanding how the average profit function of the two types of agents varied across generations. The aim was to provide insights into the temporal evolution of agent behavior within the simulation. Fig. 3.1 shows a comparison of profit function



(a) (b)

Figure 3.1: Comparison of the average profit obtained by cooperative agents (plot (a)), compared to obtained one by non-cooperatives (plot (b)).

values relative to the percentage of cooperative agents f. The horizontal axis represents the generation number or the number of groups. It is observed that as the number of cooperative agents increases, there is a linear increase in the profit function value. Notably, for $f = 0.8$ and $f = 1.0$, the average profit function exhibits similar growth patterns, ultimately converging to the same value after 10 generations. This suggests that even when the group is composed of mainly but not totally of cooperative agents, the can contribute to enhancing the profits of others. Similarly, the plot in Fig. 3.1b demonstrates that a non-cooperative strategy proves effective only when a substantial number of agents opt for it. In this case, as well, the average profit function reaches optimal values for $f = 0.2$ and $f = 0.4$, explaining why the presence of a few non-cooperative agents leads to better results.

In Fig. 3.2, a comparison of the average profit function is presented for different scenarios considering both cooperative and non-cooperative evacuees. Fig. 3.2a focuses on the situation with 2 cooperative evacuees and 8 non-cooperative evacuees, while Fig. 3.2e examines the symmetric scenario with 8 cooperative evacuees and 2 non-cooperative evacuees. A similar distinction exists in Fig. 3.2 for $f = 0.4$, represented in Fig. 3.2b, and for $f = 0.6$, depicted in Fig. 3.2d, each with 4 and 6 evacuees of each kind in two symmetric situations. These plots reveal that the av-

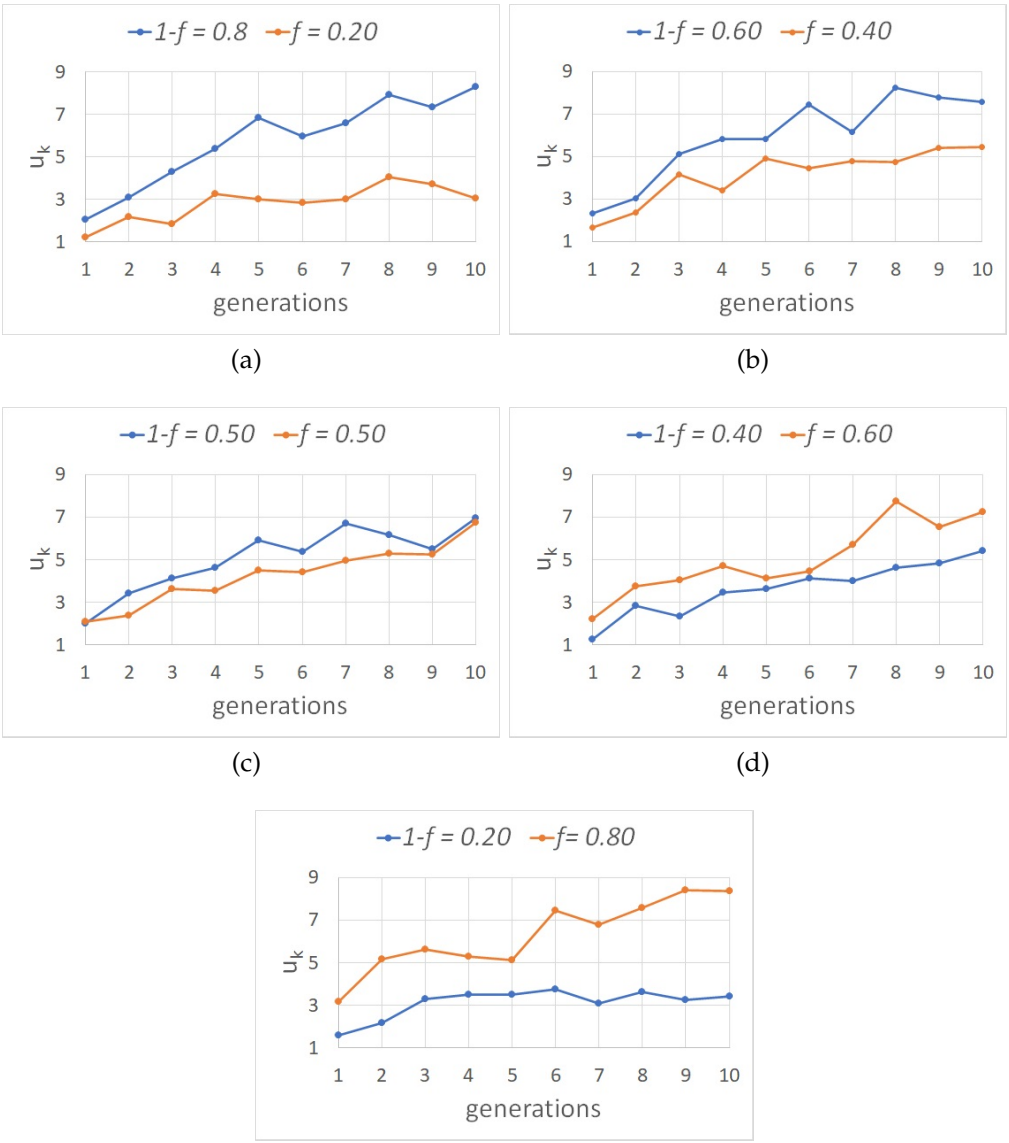


Figure 3.2: Average profit function comparison obtained by the cooperative and non-cooperative agents, at different values of f and $(1-f)$.

erage profit function tends to be higher for larger groups, specifically non-cooperative groups for $f < 0.5$ and cooperative groups for $f > 0.5$.

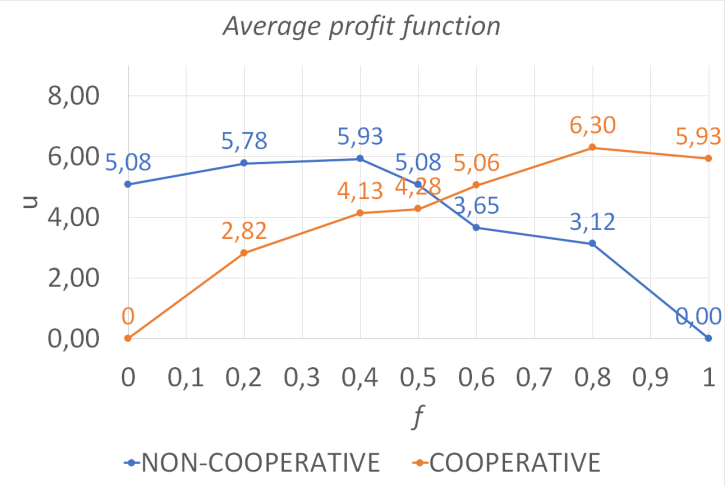


Figure 3.3: Average profit function comparison over 10 simulations and over 10 generations for cooperative and non-cooperative evacuees.

This behavior is driven by the fact that these plots are calculated for a cooperation percentage less than $f = 0.5$. However, a significant transition occurs at $f = 0.5$. In Fig. 3.2c, it can be observed that the trend of the average profit function for cooperative evacuees starts to dip below that of non-cooperative evacuees. Yet, as the generations progress, these two functions gradually converge toward the same value.

Overall type Analysis

To gain a deeper understanding of the data, a different kind of analysis was conducted to provide a comprehensive view of how the choice of cooperation percentage influenced the agents' performance. The results are depicted in Fig. 3.3 in which average values of the profit function were computed. It is evident that as the percentage of f increases, two distinct trends emerge.

For cooperative evacuees, the average profit function increases with the rise in f. Conversely, for non-cooperative evacuees, the average profit function decreases as f increases. Specifically, the average profit function for $f \geq 0.50$ surpasses that for $f \leq 0.50$, underscoring the non-equivalence of average values between two opposing and symmetric con-

figurations.
The asymmetry in the curves is a consequence of the differing dynamics between cooperative and non-cooperative agents. Furthermore, these effects are significantly influenced by the ratio of cooperative to non-cooperative agents, with higher values of the profit function (u) associated with larger values of the parameter f.

3.2.2 Conclusions

In this initial study, cooperative and competitive dynamics were analyzed within a generic evacuation process from a game theory perspective. The model consisted of a set of agents distributed across different groups aiming to reach a secure location for survival, where the shelter's capacity decreased as agents arrived. If the shelter reached full capacity, it relocated to another environment position. Agents could choose between two strategies: Non-Cooperative, disrupting a randomly selected node, or Cooperative, repairing a damaged node. The proportion of cooperative and non-cooperative agents was controlled by the cooperative factor (f), ranging from 0.0 to 1.0 in increments of 0.20. Two types of analyses, the *Generation type analysis* and the *Overall type analysis*, were conducted to examine cooperation levels' influence on agent behavior and average profit functions.

In the Generation type analysis, it was noted that as the proportion of cooperative agents (f) and the number of generations increased, the average profit function exhibited a corresponding increase. Specifically, it displayed a linear growth pattern with the increment of cooperative agents, showing similar trends for $f = 0.8$ and $f = 1.0$, eventually converging to a consistent value after 10 generations. This observation indicates that even in groups primarily composed of cooperatives but not entirely so agents, favorable average profit outcomes were achievable. Consequently, the non-cooperative behavior of a few agents could positively impact the profits of others. A detailed comparison of the average profit function across different scenarios, ranging from predominantly cooperative to predominantly non-cooperative groups, demonstrated a consistent tendency: higher average profit functions for non-cooperative groups when $f < 0.5$, and for cooperative groups when $f > 0.5$. However, a

notable transition occurred at $f = 0.5$, where cooperative evacuees displayed lower average profit functions than non-cooperative ones. Despite this, across successive generations, these functions gradually converged, suggesting the influence of temporal dynamics in aligning outcomes.

In the Overall type Analysis, two clear trends emerged when assessing the impact of the cooperation percentage (f) on agent performance. Cooperative evacuees demonstrated a consistent increase in the average profit function with higher values of f. Conversely, non-cooperative evacuees exhibited a decreasing trend in the average profit function as f increased. However, it's noteworthy that the average profit function for cooperative agents doesn't peak when $f = 1.0$, indicating that having the group entirely composed of collaborative agents doesn't necessarily maximize profit.

This observation served as the basis for subsequent analyses in the evolution of this agent-based model. Recognizing the significance of the ratio between cooperative and non-cooperative agents, further investigations were conducted to understand how the interaction between cooperative and non-cooperative dynamics could shape the complexities of social systems.

3.3 The second crowd model

The second version of the model, as documented in [20, 19, 24], was built upon the insights garnered from its predecessor. This updated version transitioned the model into a more generalized crowd simulation framework. To elaborate, it consisted of a situation that was modeled in which a population of N agents, divided into Γ groups, explores the environment and attempts to reach the destination as quickly as possible using the path with the lowest cost. In this scenario, permission is granted for all agents to reach the exit, and each group begins its exploration at regular fixed intervals. The environment is here modeled as a weighted undirected graph $G = (V, E, w)$, where V is the set of vertices, $E \subseteq V \times V$ is the set of edges, and $w\colon V \times V \to \mathbb{R}^+$ is a *weighted* function that assigns to each edge of the graph a positive cost. The *weighted* function indicates how difficult is for the agents to cross an edge. Let be $A_i = \{j \in V : (i, j) \in E\}$

the set of vertices adjacent to vertex i and $\pi^k(t) = (\pi_1, \pi_2, \ldots, \pi_t)$ a non-empty sequence of vertices, with repetitions, visited by an agent k at the time-step t, where $(\pi_i, \pi_{i+1}) \in E$ for $i = \{1, \ldots, t-1\}$. Lastly, the behavioral strategies of the agents were expanded to encompass a broader range of possibilities. The behavioral strategies for each agent have been altered as follows:

1. collaborators C: they *leave* a piece of information $\eta_{ij}(t)$ after crossing an edge (j, i) to help other agents to choose promising paths; further, they may *repair* a destroyed edge and/or an uncrossable vertex (due to the defector's action) with a probability P_e^C and P_v^C respectively;

2. defectors D: they *do not leave* any information after crossing an edge and may accidentally *destroy* an edge and/or damage a vertex just after crossing it, with probability P_e^D and P_v^D respectively. Note that a destroyed edge is no longer crossable, while a damaged node can be reached but not crossed (as if it were a wall).

In detail, collaborators primarily perform actions that benefit all other agents in reaching the exit point as swiftly as possible. It can be assumed that they traverse edges and vertices cautiously, taking care not only to avoid damaging them but also to fix them if they are damaged. Additionally, they leave information about edge costs ($\eta_{ij}(t)$) for other agents to use when making decisions. To depict a real-world situation, this information could be imagined as written notes, directional signs, color markers, or any other form of hints that can be left behind. On the other hand, defectors predominantly behave hastily, engaging in actions that have the potential to disrupt the environment. After traversing a node or an edge, they may inadvertently damage it, reducing the chances of other agents successfully exploring the environment. This behavior may not only affect collaborative agents but also have consequences for the defectors themselves, particularly if the disrupted path is critical for reaching the exit. This behavior of defectors can be attributed to panic, fear, stress, and negligence, similar to situations where agents, like real humans, may act without full awareness of their actions. They simply

try to find a viable path by following others and without communicating their discoveries to the rest of the group.

Similarly to the first model, the cooperative and defector strategies in this second model were implemented straightforwardly. However, a key distinction from the previous model was the incorporation of user-defined probabilities for destruction and repair within the behaviors of both collaborators and defectors. This addition allowed for adjusting the intensity of these behaviors across the agents, emphasizing the significance of considering a diverse crowd composition encompassing individuals with varied behavioral tendencies.

Three evaluation metrics were simultaneously compared to understand how collaborative and/or defector strategies affect the two types of agents and the crowd's overall behavior. This included the number of agents reaching the exit (Success Rate or SR), exit times, and path costs for reaching the exit. The analysis involved normalizing exit times and path costs relative to the Success Rate. Various kinds of analyses were conducted, involving different parameter values and scenarios:

- **Type Analysis:** This analysis focused on comparing the performance of the two types of agents, namely collaborators and defectors. It aimed to understand how they influenced each other in different situations.

- **Group Analysis:** In this analysis, the performance of the groups into which the crowd was divided was compared. The goal was to analyze the behavior of different groups.

- **Overall Analysis:** This broader analysis considered the performance of the entire crowd as a whole. It aimed to provide a comprehensive view of how the collective behavior of all agents, both collaborators and defectors, influenced the outcomes.

- **Sensitivity Analysis:** A sensitivity analysis was conducted to evaluate the robustness of the previous analyses with respect to specific parameters. This involved varying a particular parameter and examining how it affected the results obtained. Such an analysis helps in understanding the degree to which the model's outcomes are influenced by changes in specific factors.

3.3.1 Experiments and results

For all the analyses (except for the sensitivity one), a total of $N = 1000$ agents were considered, grouped into $\Gamma = 10$ distinct groups, with each group comprising 100 agents. Within each group, the composition of agents was determined by a user-defined parameter denoted as the "collaborative factor," denoted as $f \in [0, 1]$. Depending on the value of f, a group could consist of both collaborative and defector agents, or of just one type. Specifically, if $f = 0.0$, only groups composed entirely of defectors were analyzed. If $f = 0.5$, each group was equally split between collaborative and defector agents, while if $f = 1.0$, only groups composed entirely of collaborators were examined. In terms of the exploration timeline, except for the initial group, which logically started at time $T_e = 0$, each subsequent group commenced its exploration at a specific time $T_e = |V|$ after the previous group. Generally, the i-th group initiated its exploration at $T_e \times (i - 1)$. A time limit, T_{max}, was imposed on all agents, calculated as follows: $T_{max} = 2 \times \Gamma \times T_e$, where Γ represents the number of groups, and 2 is a fixed parameter. The time frame within which each agent had to reach the exit extended from the commencement of its exploration to the overall maximum time T_{max} (i.e., $T_{max} - (T_e \times (i - 1))$). This time allocation mechanism meant that the initial groups had more time to explore the environment compared to later groups. The model permitted groups to commence their exploration even if agents from earlier groups were still present in the environment. Consequently, agents within the same group could exit at different times, always within their designated time range. Agents in the initial group consistently had more time available to reach the exit. Regarding the trace release and deterioration process, the trace left along the path decayed over time, with the global updating rule applied every $T_d = 50$ ticks. The evaporation rate for the trace was set to $\rho = 0.10$. Initially, the trace was set to $\tau_{ij}(t = 0) = 1.0$ for all edges $(i, j) \in E$. The parameters α and β, which regulated the importance of the trace and desirability in the transition probability equation, were both set to 1.0. Concerning the destruction and repair probabilities, the probabilities of a vertex and/or an edge being destroyed were specifically set to $P_e^C = P_e^D = 0.02$ for edge destruction and $P_v^C = P_v^D = 0.02$ for both types of agents. Lastly, the

analysis involved running a total of 10 independent simulations for each value of f, ranging from 0.0 to 1.0 in steps of 0.1, using, depending on the analysis made, the following scenarios of increasing complexity:

- scenario A1, with $|V| = 100$; $|E| = 153$; and generated by setting $p_1 = 0.6$, $p_2 = 0.0$;
- scenario A2, with $|V| = 100$; $|E| = 213$; and generated by setting $p_1 = 0.6$, $p_2 = 0.2$;
- scenario B1, with $|V| = 225$; $|E| = 348$; and generated by setting $p_1 = 0.6$, $p_2 = 0.0$;
- scenario B2, with $|V| = 225$; $|E| = 495$; and generated by setting $p_1 = 0.6$, $p_2 = 0.2$;
- scenario C1, with $|V| = 400$; $|E| = 642$; and generated by setting $p_1 = 0.6$, $p_2 = 0.0$;
- scenario C2, with $|V| = 400$; $|E| = 889$; and generated by setting $p_1 = 0.6$, $p_2 = 0.2$.

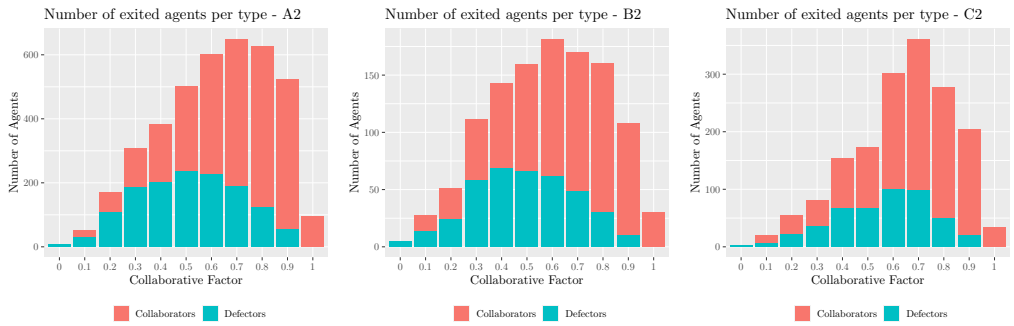
The two parameters p_1 and p_e control a node's connectedness with its neighbors. In detail, $0 \leq p_1 \leq 1$ represents the probability of creating horizontal and vertical edges, and $0 \leq p_2 \leq 1$ represents the probability of creating oblique edges. In this way, Type 1 scenarios featured solely vertical and horizontal edges, while Type 2 scenarios incorporated oblique edges.

Type Analysis

For the type analysis, the following scenarios $A2$, $B2$, and $C2$ were considered. Fig. 3.4 displays the absolute agent counts for scenarios $A2$, $B2$, and $C2$ over various collaborative factor (f) values. Both collaborative and defector agents are included in the analysis, with coral indicating collaborative agents and turquoise representing defectors. When assessing the collective behavior, it becomes apparent that crowd performance improves as f increases, reaching a peak and subsequently declining in all three scenarios.

The highest number of exited agents is achieved at different f values for each scenario: in scenario $A2$, the peak occurs at $f = 0.7$, with nearly 650 agents exiting (see Fig. 3.4a); in scenario $B2$, the maximum is at $f = 0.6$, with almost 180 agents exiting (see Fig. 3.4b); and in scenario $C2$, the peak occurs at $f = 0.7$, with approximately 360 agents exiting (see Fig. 3.4c). Analyzing these peak values suggests that crowd performance is best in scenario $A2$, followed by scenario $C2$, and lastly, scenario $B2$. This variation in performance could be attributed to the complexity of the scenario and the presence of specific node and edge configurations.

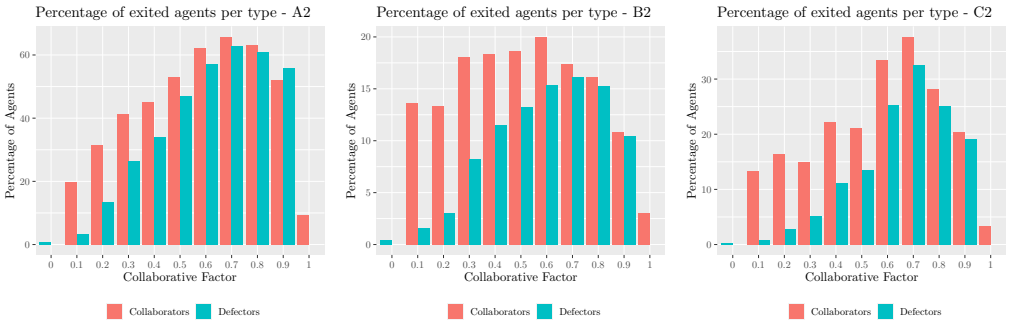
When $f = 1.0$, indicating the presence of only collaborative agents, overall performance is at its worst. Similarly, the same unfavorable performance results are observed when $f = 0.0$, signifying the presence of only defector agents. This outcome is unsurprising, as defector agents obstruct and block parts of their pathways with a certain probability, rendering them physically incapable of progressing toward the exit. Fig. 3.5



(a) (b) (c)

Figure 3.4: Absolute counts for (a) scenario A2, (b) scenario B2, and (c) scenario C2.

presents the relative counts, which differ from the previous plots as they illustrate the percentage of agents, both collaborators and defectors, that successfully reach the exit. This percentage is calculated based on the total number of agents, denoted as N (where $N = 1000$). In this context, it is evident that agent performance improves as f increases, but variations exist depending on the agent type and the specific scenario under consideration.



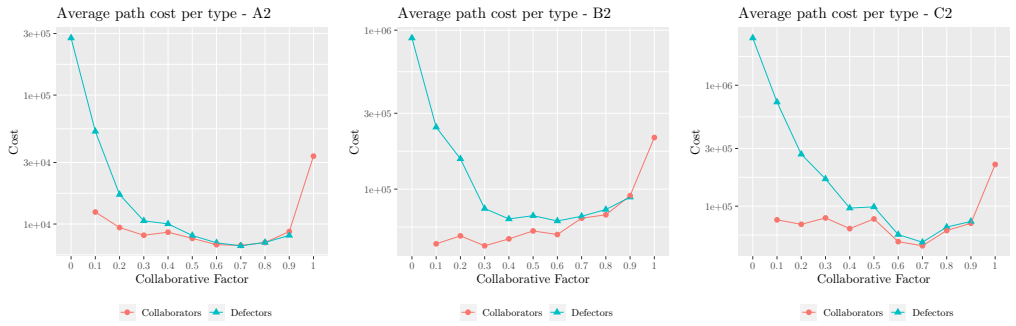
(a) (b) (c)

Figure 3.5: Relative count for (a) scenario A2, (b) scenario B2 and (c) scenario C2.

In scenario $A2$ (Fig. 3.5a), both collaborators and defectors achieve their highest exit percentages at $f = 0.7$. Beyond this threshold, their performance declines, though defectors maintain relatively good performance even when outnumbered ($f = 0.9$). Collaborators improve as f increases, peaking before dropping at $f = 1.0$, where only 10% of them reach the exit. Similar trends apply to defectors, but their performance drop is less steep. Scenario $B2$ (Fig. 3.5b) shows peak exit percentages at $f = 0.6$ for collaborators (20%) and $f = 0.7$ for defectors (16%). Collaborators consistently outperform defectors, even when they are scarce ($f < 0.5$). Collaborator performance decreases as f rises, while defectors maintain decent performance. In scenario $C2$ (Fig. 3.5c), both collaborators and defectors achieve their highest exit percentages at $f = 0.7$, with collaborator performance at almost 37% and defectors at almost 32%. Above this point, exit percentages decline with increasing f. Scenario $C2$ generally has lower exit percentages than scenario $A2$ but higher than scenario $B2$. Collaborators consistently outperform defectors when they are in the minority ($f < 0.5$). Defectors improve performance when outnumbered ($f > 0.5$), but it decreases beyond that point. In all scenarios, when $f = 1.0$, indicating a fully collaborative crowd," the percentage of collaborators reaching the exit is significantly lower than for lower f values.

In conclusion, the results (Fig. 3.4) indicate that not all 1000 agents

reach the exit. Performance varies depending on the scenario complexity and agents' limited time. Having defectors in mixed crowds ($0.0 < f < 1.0$) increases exit percentages compared to fully collaborative crowds ($f = 1.0$). Fig. 3.5 shows that when collaborators outnumber defectors ($f > 0.5$), defectors leverage collaborators' information to achieve high exit percentages.

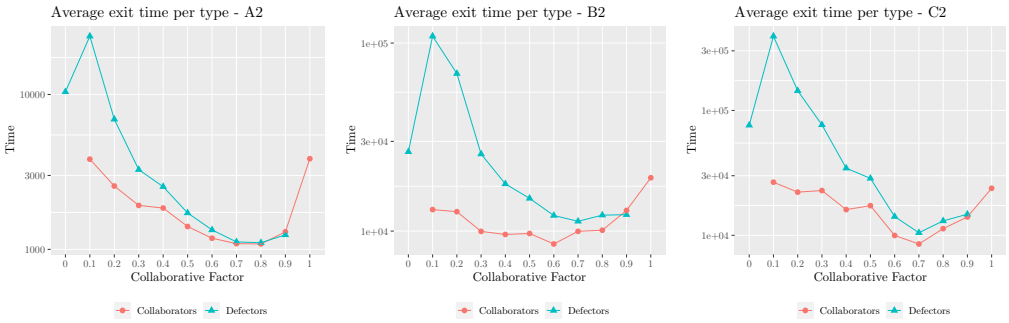


(a) (b) (c)

Figure 3.6: The path cost for (a) scenario A2, (b) scenario B2 and (c) scenario C2.

In Fig. 3.6 and Fig. 3.7, both path costs and exit times are normalized with respect to the agents' success rate (SR). This normalization makes these evaluation metrics more statistically meaningful, as it considers both cost and speed in conjunction with the number of agents reaching the exit. Regarding path costs in scenario $A2$ (Fig. 3.6a), collaborators and defectors exhibit opposing trends as f varies. Defectors significantly reduce their path costs as f increases, reaching a minimum at $f = 0.7$. Collaborators gradually reduce their path costs with increasing f, achieving the minimum at $f = 0.7$ but experiencing a sharp cost increase at $f = 1.0$. In scenario $B2$ (Fig. 3.6b), defectors' path costs increase with higher f, while collaborators' costs worsen. Defectors notably improve their performance with a cost minimum at $f = 0.6$, whereas collaborators find an optimal path at $f = 0.3$ but worsen as f increases, with the worst performance at $f = 1.0$. Scenario $C2$ (Fig. 3.6c) follows a similar pattern. Defectors and collaborators both find their optimal path at $f = 0.7$, with defectors continuing to improve with higher f. Collaborators gradually

improve but experience a sudden cost increase at $f = 1.0$.



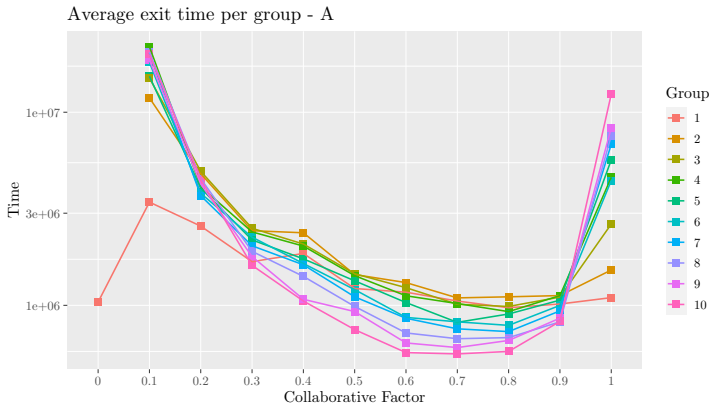
(a) (b) (c)

Figure 3.7: The exit time for (a) scenario A2, (b) scenario B2 and (c) scenario C2.

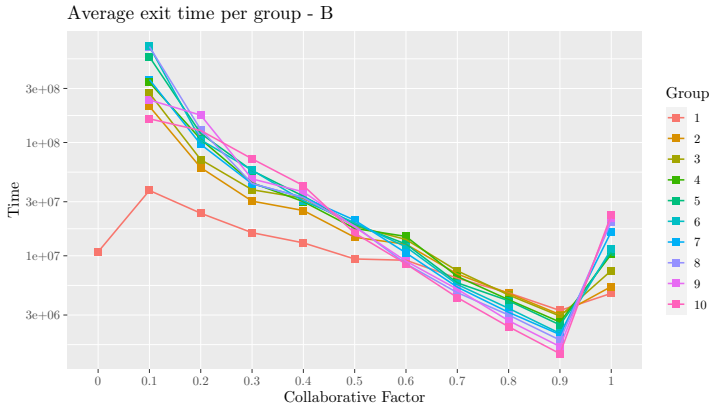
In scenario $A2$ (Fig. 3.7a), the exit times for defectors are initially slower at $f = 0.0$ compared to $f = 0.1$. However, as f increases beyond 0.1, defectors improve their exit times, reaching the quickest time at $f = 0.8$. A slight increase in exit time occurs at $f = 0.9$. Collaborators also gradually improve their exit times, achieving the minimum at $f = 0.8$, coinciding with defectors. Strangely, the worst exit times for both collaborators and defectors occur at $f = 1.0$, when the crowd consists solely of collaborators. In scenario $B2$ (Fig. 3.7b), similar patterns emerge. Defectors perform poorly at $f = 0.1$ but improve their exit times beyond this point, reaching the minimum at $f = 0.7$. Collaborators show improvement until $f = 0.6$, after which their exit times worsen.
Interestingly, the worst exit times for both groups occur at $f = 1.0$. Scenario $C2$ (Fig. 3.7c) follows a similar trend. Both collaborators and defectors achieve their minimum exit times at $f = 0.7$. Collaborators exit slowly at $f = 1.0$, but unlike the previous scenarios, their performance at this value is better than at $f = 0.1$.

Group Analysis

For the group analysis, the following scenarios $A2$ and $B1$ were considered.



(a)



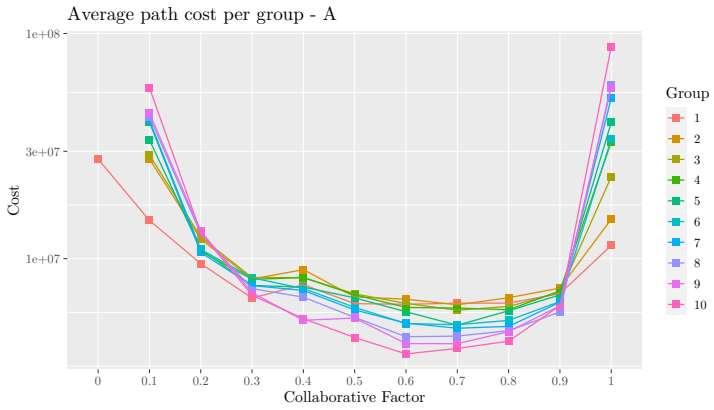
(b)

Figure 3.8: The exit time for (a) scenario A and (b) scenario B.

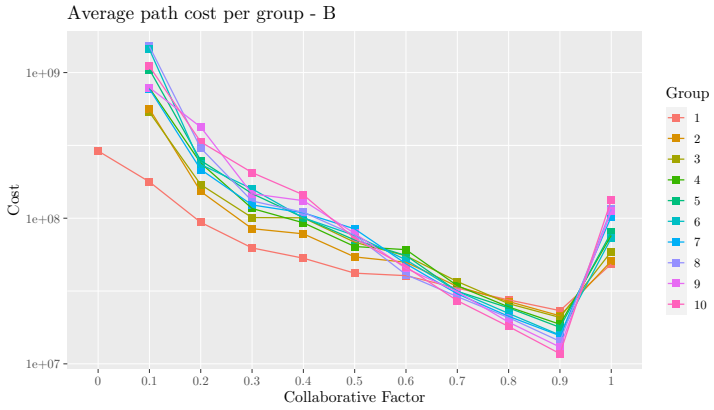
The exit time (Fig. 3.8) and path cost (Fig. 3.9) plots have been normalized with respect to the group success rate, that is the percentage of agents in a group, which successfully reach the exit point. The better the agents perform, the lower these values are. In both scenarios, exit time improves as the collaborative factor increases, suggesting that more cooperation leads to faster exits. However, this trend breaks at $f = 1.0$, where all agents collaborate, resulting in worse performance for each group compared to previous collaboration levels. Groups are represented by differently colored lines, and it appears that exit time de-

creases as the group number increases. This suggests that even groups leaving later somehow use information left by those who evacuated earlier, even if they have less time. For example, Group 1 performs poorly at low collaboration levels but excels at high collaboration, indicating that this group can better use path information, especially when most agents are collaborative. Don't be misled by the fact that the same group performs well even at the lowest collaboration level (f). Later, it will be demonstrated that at this level, only a few agents manage to find the exit. Considering both exit time and the number of agents who successfully exit, it can concluded that this result is not particularly significant. Conversely, examining Group 10 in both scenarios reveals improved performance with increasing collaboration (f), except when collaboration is at its maximum ($f = 1.0$). This suggests that the final group can utilize path information more effectively when the crowd is predominantly, but not entirely, composed of collaborative agents. Similar conclusions can be extended to the path cost depicted in Fig. 3.9, which decreases in relation to both the collaborative factor and group number. This holds true for every f value except for $f = 1.0$. It indicates, as mentioned earlier, that heightened collaboration among agents leads to better pathfinding, but if collaboration is absolute, it appears to be ineffective.

The heat maps in Fig. 3.10 illustrate the number of agents reaching the exit, normalized according to the exit time available for each group. They show the rate at which agents are evacuated per unit of time (or at each tick). A higher value here indicates better agent performance. Interestingly, a pattern opposite in value to the trends observed in exit time and path cost is evident. It appears that the number of agents exiting increases with the collaborative factor, except, once again, for $f = 1.0$, where only a few agents manage to reach the exit. Surprisingly, the agents' performance isn't better when all of them collaborate; instead, it seems more effective when some act as defectors. Additionally, this quantity appears to increase concerning the group number, suggesting that later groups benefit from the actions of earlier groups, particularly in scenario A.



(a)



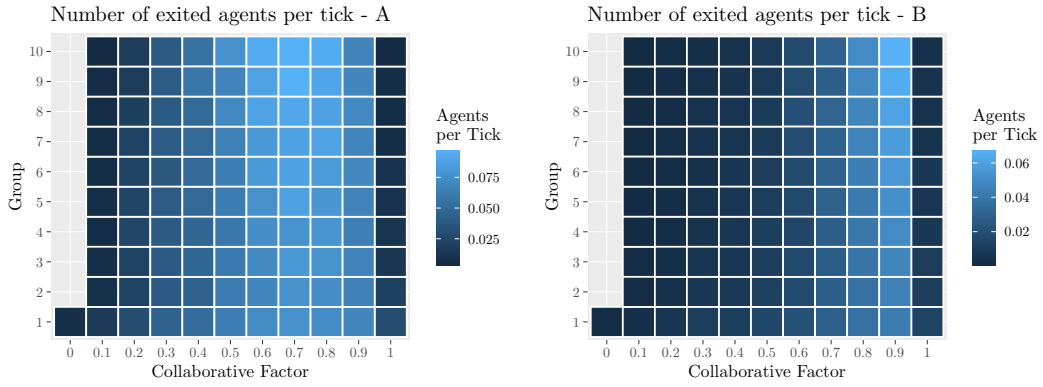
(b)

Figure 3.9: The path cost for (a) scenario A and (b) scenario B.

Overall Analysis

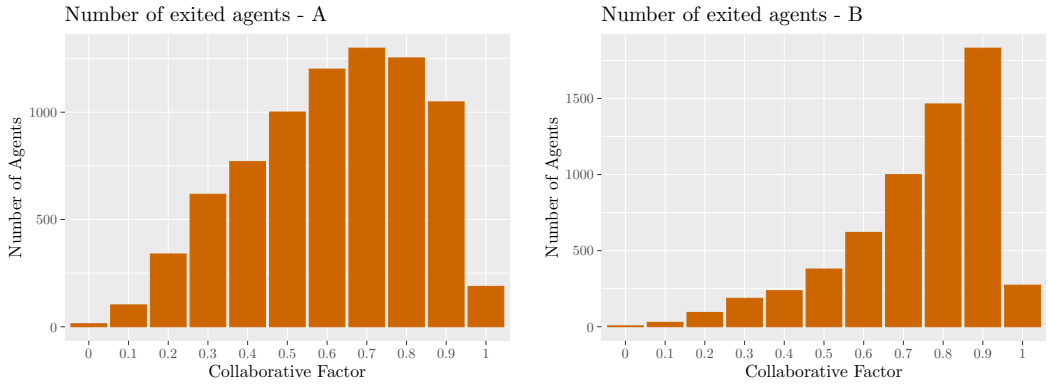
Also for the group analysis, the following scenarios $A2$ and $B1$ were considered.

In Fig. 3.11, the total number of exited agents is shown for both scenario A and scenario B. Regardless of the group number, similar conclusions can be drawn as mentioned earlier: the number of agents reaching the exit increases with the collaborative factor, providing a clearer view of the collective behavior of the simulated crowd. The highest number of exited agents occurs at $f = 0.7$ for scenario A and at $f = 0.9$ for scenario



(a) (b)

Figure 3.10: The number of exited agents per tick for scenarios A and B.



(a) (b)

Figure 3.11: The number of exited agents for scenarios A and B.

B.

Analyzing the overall trend of the metrics used (number of agents reaching the exit, path cost, and exit time), it becomes apparent that the agents' best performances don't occur when the entire group is entirely collaborative but rather when some act as defectors. Essentially, the crowd seems to perform optimally when there's a mix of collaborative and non-collaborative agents, suggesting the effectiveness of a diverse

strategy among the agents.

Sensitivity Analysis

In the context of the sensitivity analysis, all six scenarios, namely $A1$, $A2$, $B1$, $B2$, $C1$, and $C2$, were taken into consideration. The comparison of collaborators and defectors was conducted across various combinations of K, α, and β. Each figure is composed of two lines representing trace values of $K = 0.001$ and $K = 0.1$, and three columns representing different configurations of α and β parameters. Each plot provides insights into the results corresponding to a specific K value and a particular α and β configuration. The x-axes depict the collaborative factor f, while the y-axes showcase the evaluated metric. The discussion will refer to terms like $\alpha < \beta$ for $\alpha = 0.5$ and $\beta = 1.0$, $\alpha > \beta$ for $\alpha = 1.0$ and $\beta = 0.5$, and $\alpha = \beta$ for $\alpha = \beta = 1.0$. The condition when $K = 0.001$ will be denoted as the "low-trace condition," and when $K = 0.1$, it will be referred to as the "high-trace condition". The purpose of the parameter K is to represent scenarios where information about other agents' actions is either more visible (high-trace) or less visible (low-trace). For each K value, if $\alpha < \beta$, agents prioritize the trace τ_{ij} over the information η_{ij} in selecting the next path. Conversely, if $\alpha > \beta$, they assign more weight to the information. If $\alpha = \beta$, equal importance is given to both the trace and the information.

The *absolute count* bar charts represent how many agents, of both types, have reached the exit. The defectors' bars are under the collaborators' ones. In scenario A1, Fig. 3.12, we have that:

- when $K = 0.001$, i.e. when the trace is low, collaborators and defectors reach the exit more or less equally, regardless of the values configurations of α and β. In particular, defectors exit more as the collaborative factor increases and reach the maximum when $f = 0.6$. Collaborators exit more as the collaborative factor increases, reaching the maximum when $f = 1.0$;

- when $K = 0.1$, defectors exit more as f increases and reach the maximum when $f = 0.6$ for $\alpha < \beta$ and $\alpha > \beta$, and when $f = 0.7$ for $\alpha = \beta$. Collaborators exit more when $f = 1.0$ only if $\alpha < \beta$. If $\alpha > \beta$ and $\alpha = \beta$ they reach the maximum when $f = 0.9$.

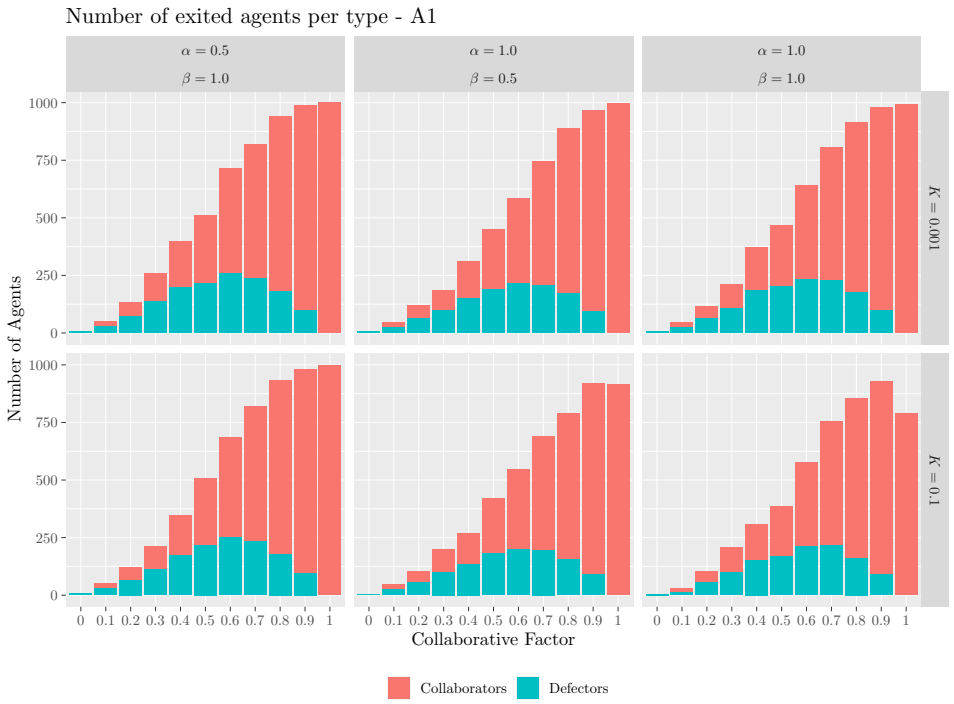


Figure 3.12: Number of exited agents per type for scenario A1.

Inspecting scenario A2, which is displayed in Fig. 3.13, it is possible to assert that:

- when $K = 0.001$, defectors exit more when $f = 0.5$ if $\alpha < \beta$ and $\alpha = \beta$. If $\alpha > \beta$ they reach the maximum when $f = 0.6$. Collaborators exit more as the collaborative factor increases, reaching the maximum when $f = 1.0$ but only when $\alpha < \beta$ and $\alpha > \beta$. If $\alpha = \beta$ they exit more when $f = 0.9$;
- when $K = 0.1$, the behavior of both defectors and collaborators is similar to the previous case. We have, indeed, that defectors exit more for the same values configurations. Collaborators reach the maximum when $f = 0.9$ for $\alpha < \beta$, when $f = 0.7$ for $\alpha > \beta$, and $\alpha = \beta$.

In scenario A1 (Fig. 3.12), the number of outgoing agents, both collaborators and defectors, decreases from the top left plot to the bottom right

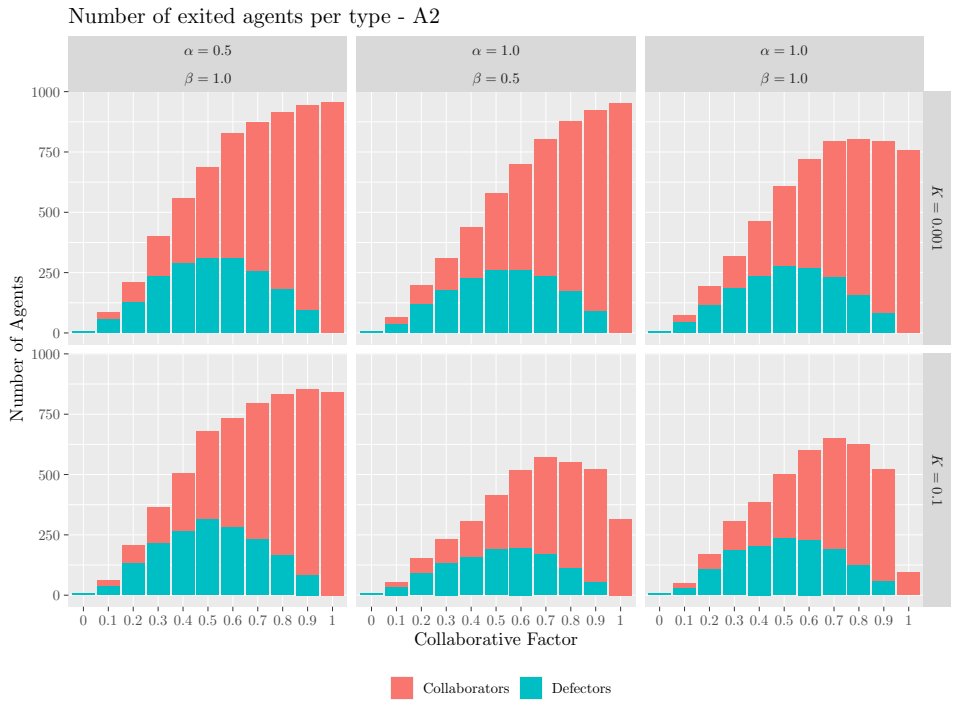


Figure 3.13: Number of exited agents per type for scenario A2.

plot—transitioning from $K = 0.001$ and $\alpha < \beta$ to $K = 0.1$ and $\alpha = \beta$. For $K = 0.001$, when the trace is low, collaborators and defectors perform almost identically across different α and β values. This suggests that, with a low trace, the agents exit in similar quantities, regardless of whether they give more weight to the discovered information η ($\alpha < \beta$), to the trace ($\alpha > \beta$), or equal weight to both parameters ($\alpha = \beta$). Specifically, collaborators show higher numbers when $f = 1.0$, while defectors excel at $f = 0.6$.

This trend becomes more pronounced with a higher trace value, i.e., $K = 0.1$. In this case, the number of outgoing agents decreases from the left plot to the right, indicating that more agents leave when they prioritize the discovered information η ($\alpha < \beta$) compared to giving equal weight to trace and information ($\alpha = \beta$). Collaborators shift from a peak at $f = 1.0$ when $\alpha < \beta$ to a maximum at $f = 0.9$ when $\alpha > \beta$ and $\alpha = \beta$. This implies that the presence of defectors is beneficial for the entire

crowd only under specific conditions—when the trace is high, and agents assign equal weight to trace and information. Under these conditions, more agents exit than when the crowd consists entirely of collaborative agents.

In scenario A2 (Fig. 3.13), the agents exhibit a similar behavior to that discussed for A1, with slightly worse performance. Again, the number of exited agents decreases from the upper left plot to the lower right plot, signifying a deterioration in performance when transitioning from low trace with greater weight to information to high trace with equal weight to trace and information.

It's crucial to note that this trend is not exclusive to scenarios A1 and A2 but applies to the remaining scenarios (B1, B2, C1, and C2) and all evaluation metrics (relative counts, path costs, and exit times). Therefore, the discussion will focus on scenarios A1 and A2 for each metric, as similar conclusions can be drawn for the others.

Indeed, in scenario B1, Fig. 3.14, we have that:

- when $K = 0.001$, i.e. in a low-trace condition, defectors exit more when $f = 0.7$ for $\alpha < \beta$ and $\alpha > \beta$. When $\alpha = \beta$ they reach instead the maximum peak when $f = 0.8$. Regarding the collaborators, instead, if $\alpha < \beta$ more come out when $f = 1.0$, while for $\alpha > \beta$ and especially $\alpha = \beta$ they exit more when $f = 0.9$;
- when $K = 0.1$, we have that defectors exit more for $f = 0.9$ if $\alpha < \beta$ and $\alpha = \beta$. If $\alpha > \beta$, they exit more for $f = 0.8$. Collaborators reach the maximum value for $\alpha < \beta$ when $f = 1.0$; on the other hand, for $\alpha > \beta$ and $\alpha = \beta$ they reach the maximum peak for $f = 0.9$.

In scenario B2, Fig. 3.15, we have that:

- setting $K = 0.001$, defectors exit more when $f = 0.6$ independently on the configuration values of α and β; while collaborators exit more for $f = 1.0$ when $\alpha < \beta$ and $\alpha > \beta$, and for $f = 0.8$ when $\alpha = \beta$;
- setting $K = 0.1$, defectors exit more for $f = 0.6$ when $\alpha < \beta$ and $\alpha > \beta$, and $f = 0.4$ when $\alpha = \beta$. Collaborators instead reach the maximum number of exited agents at $f = 1.0$ only when $\alpha < \beta$; when $\alpha > \beta$ and $\alpha = \beta$, they reach the maximum values for $f = 0.8$ and for $f = 0.6$ respectively.

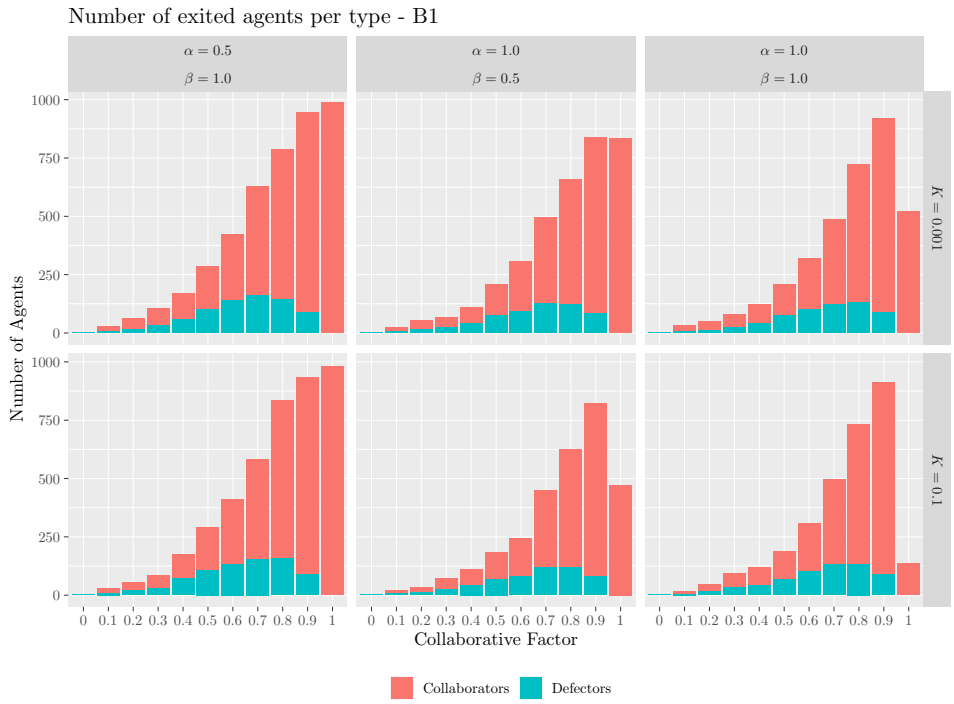


Figure 3.14: Number of exited agents per type for scenario B1.

Inspecting the outcomes obtained in scenario C1, displayed in Fig. 3.16, is possible to assert that:

- when $K = 0.001$, defectors exit more when $f = 0.7$ for $\alpha < \beta$ and $\alpha = \beta$, while for $f = 0.8$ when $\alpha > \beta$. Collaborators exit more for $f = 1.0$ only when $\alpha < \beta$; instead, for $\alpha > \beta$ and $\alpha = \beta$ they exit more when $f = 0.9$;

- when $K = 0.1$, defectors exit more for $f = 0.7$ if $\alpha < \beta$ and $\alpha > \beta$. If $\alpha = \beta$, they exit more for $f = 0.6$. Collaborators reach the maximum when $f = 1.0$ but only for $\alpha < \beta$. If $\alpha > \beta$ and $\alpha = \beta$, they reach the maximum respectively for $f = 0.9$ and for $f = 0.8$.

From the outcomes in scenario C2, reported in Fig. 3.17, we have that:

- when $K = 0.001$, which means the trace is low, defectors exit more when $f = 0.7$ if $\alpha < \beta$ and $\alpha > \beta$, and when $f = 0.6$ if $\alpha = \beta$.

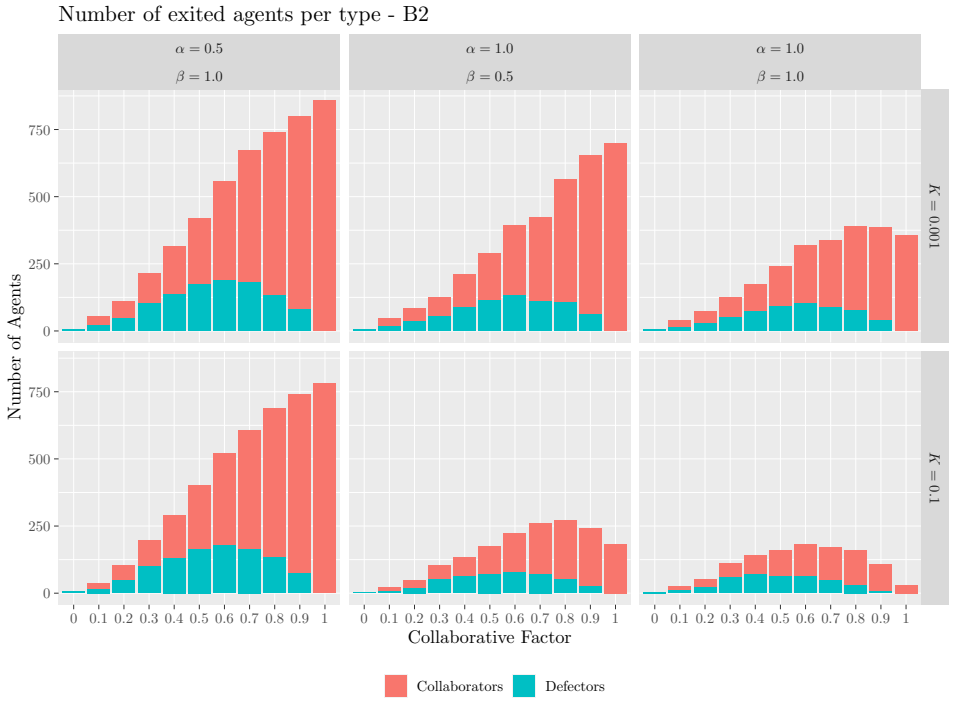


Figure 3.15: Number of exited agents per type for scenario B2.

Collaborators, instead, exit more when $f = 1.0$ if $\alpha < \beta$; $f = 0.9$ when $\alpha > \beta$; and $f = 0.8$ when $\alpha = \beta$;

- when $K = 0.1$, defectors exit more for $f = 0.6$ if $\alpha < \beta$ and $\alpha = \beta$. If $\alpha > \beta$, they exit more for $f = 0.7$. Collaborators reach the maximum when $f = 1.0$ but only for $\alpha < \beta$. If $\alpha > \beta$ and $\alpha = \beta$, they exit more $f = 0.7$.

In real-life scenarios, this finding implies that there may be situations where incorporating individuals who deviate from collaborative behavior (defectors) can improve the overall exit process. These conditions may arise when there is a significant presence of trace (e.g., clear evacuation signage, marked pathways) and when individuals appropriately balance the information provided by the trace with other available information (such as instructions from authorities or real-time updates). The observed trends in the number of outgoing agents, both collaborators and defec-

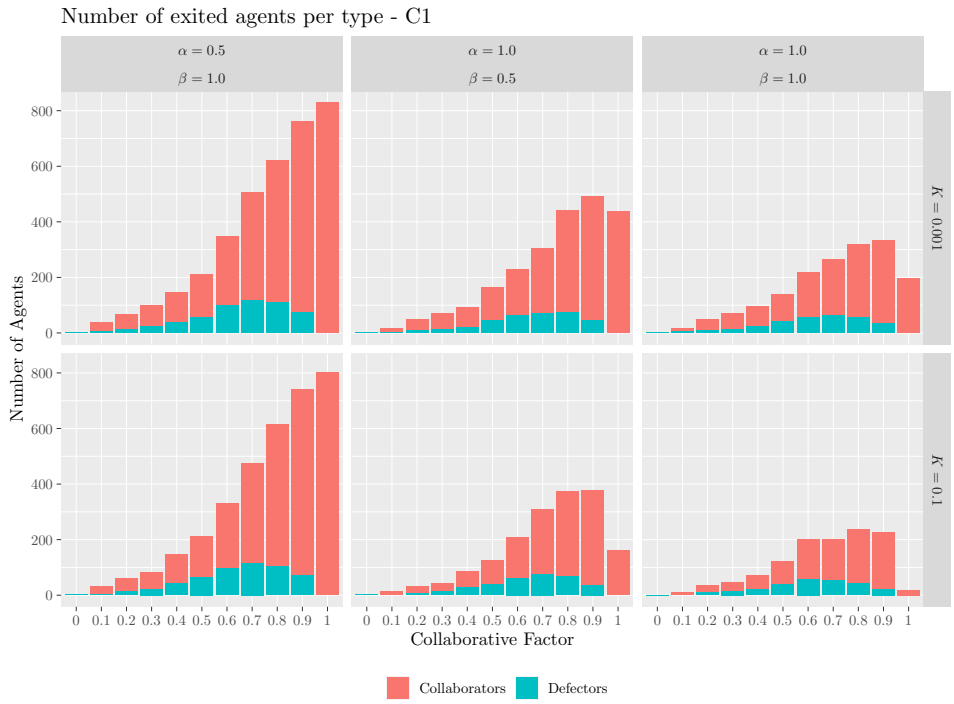


Figure 3.16: Number of exited agents per type for scenario C1.

tors, provide insights into the behavior of individuals within crowds. The decrease in the number of outgoing agents as the trace value increases and as the weight on information becomes more or equal to the trace suggests that crowd behavior is influenced by the interplay between accumulated trace and the information individuals discover.

The *relative count* plots represent the percentage of agents, of both types, that have reached the exit, with respect to their total number. Let be N_e^C (N_e^D) the number of exited collaborators (defectors), we define the *Exit Rate* (ER) as follows:

$$ER = N_e^C / N_C \quad (N_e^D / N_D), \tag{3.3}$$

where N_C (N_D) is the total number of collaborative (defector) agents. Unlike the absolute count plots, the relative ones give a piece of more detailed information. They can be considered as the exit rate of the agents.

In scenario A1, Fig. 3.18, we have that:

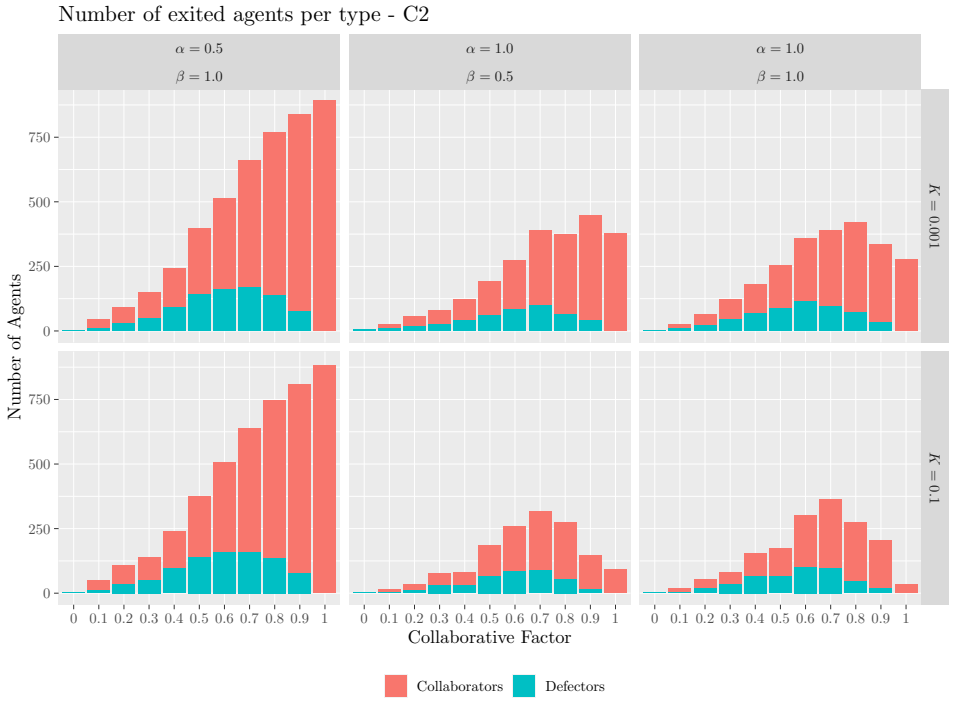


Figure 3.17: Number of exited agents per type for scenario C2.

- when $K = 0.001$, i.e. when the trace is low, the ER of both defectors and collaborators increases with f, independently of the values configurations of α and β, and is maximum for $f = 0.9$ for the former, and for $f = 1.0$ for the latter;

- when $K = 0.1$, the ER of defectors is, as in the previous case, the maximum for $f = 0.9$ independently of the values configurations of α and β. The ER of collaborators is maximum for $f = 1.0$ if $\alpha < \beta$, whilst for $\alpha > \beta$ and $\alpha = \beta$ it is $f = 0.9$.

In scenario A2, Fig. 3.19, we have that:

- when $K = 0.001$, the ER of defectors increases with f, independently of the values configurations of α and β and is maximum for $f = 0.9$. The ER of collaborators is maximum when $f = 1.0$ only if $\alpha < \beta$ and $\alpha > \beta$. If $\alpha = \beta$ the maximum is for $f = 0.9$;

Figure 3.18: Percentage of exited agents per type for scenario A1.

- when $K = 0.1$, the ER of defectors is maximum for $f = 0.9$ if $\alpha < \beta$; otherwise for $f = 0.7$ if $\alpha > \beta$ and $\alpha = \beta$. The ER of collaborators is maximum at $f = 1.0$ if $\alpha < \beta$, and at $f = 0.7$ if $\alpha > \beta$ and if $\alpha = \beta$.

In the overall, in scenario A1 (Fig. 3.18) the exit rate of all agents decreases moving from the top left plot to the bottom right plot. Therefore, moving from a configuration with $K = 0.001$ and $\alpha < \beta$ to a configuration with $K = 0.1$ and $\alpha = \beta$. As one has seen for the absolute count plots, for $K = 0.001$, i.e. in a low-trace condition, the performance of both collaborators and defectors is almost identical regardless of the values of α and β. It follows that when the trace value is low, the *ER* of both types of agents is more or less the same regardless of whether, during their decision-making process, they give more weight to the discovered information η ($\alpha < \beta$), or to the trace ($\alpha > \beta$) or equal weight to both parameters ($\alpha = \beta$). In particular, the percentage of outgoing collaborators is 100% when $f = 1.0$, while the percentage of the defectors is slightly less with

Figure 3.19: Percentage of exited agents per type for scenario A2.

respect to one of the collaborators. Also for the exit rate, this behavior is more pronounced when the trace value is higher, i.e. when $K = 0.1$ and it can be seen that the exit rate of both types of agents decreases moving from the plot on the left to the one on the right. Therefore the percentage of outgoing agents is higher if, during their decision-making process, they give more weight to the discovered information η ($\alpha < \beta$), than the same weight to information and trace ($\alpha = \beta$). In this case, indeed, collaborators go from a maximum exit rate of agents reached for $f = 1.0$ when $\alpha < \beta$ to a maximum of outgoing agents reached for $f = 0.9$ when $\alpha > \beta$ and $\alpha = \beta$. As asserted before, the presence of defectors is useful for the entire crowd only under certain conditions, i.e. when the trace released is high and, at the same time, the agents give equal weight to the trace and to the information because the exit rate of the agents is higher than when the crowd is composed entirely of collaborative agents. In scenario A2 (Fig. 3.19), the behavior of the agents is very similar to that

discussed for A1, with slightly worse performance than the latter. Again, the exit rate of both types of agents decreases when moving from the upper left plot to the lower right plot. Therefore, the agents' performance deteriorates when one passes from a condition of low trace and greater weight to information to a condition of high trace and equal weight to trace and information.

Figure 3.20: Percentage of exited agents per type for scenario B1.

In scenario B1, Fig. 3.20, we have that:

- when $K = 0.001$, the ER of defectors increases with f, independently of the values configurations of α and β, and is maximum for $f = 0.9$. The ER of collaborators is maximum when $f = 1.0$ if $\alpha < \beta$, whilst for $\alpha > \beta$ and $\alpha = \beta$ it is when $f = 0.9$;
- when $K = 0.1$, we have the same results as above.

Figure 3.21: Percentage of exited agents per type for scenario B2.

Also analyzing the outcomes in scenario B2, reported Fig. 3.21, we have that:

- when $K = 0.001$, i.e. when the trace is low, the ER of defectors increases with f, independently of the values configurations of α and β, and is maximum for $f = 0.9$. The ER of collaborators is maximum when $f = 1.0$ if $\alpha < \beta$ and if $\alpha > \beta$. If $\alpha = \beta$ the maximum is for $f = 0.8$;

- when $K = 0.1$, the ER of defectors is maximum when $f = 0.9$ if $\alpha < \beta$, when $f = 0.8$ if $\alpha > \beta$ and when $f = 0.7$ when $\alpha = \beta$. The ER of collaborators, is maximum when $f = 1.0$ if $\alpha < \beta$, when $f = 0.8$ if $\alpha > \beta$ and when $f = 0.6$ when $\alpha = \beta$

From scenario C1, displayed in Fig. 3.22, it is possible to observe that:

- when $K = 0.001$, the ER of defectors also here increases with f,

Figure 3.22: Percentage of exited agents per type for scenario C1.

independently of the values configurations of α and β, and is maximum for $f = 0.9$. The ER of collaborators is maximum for $f = 1.0$ when $\alpha < \beta$, and for $f = 0.9$ when $\alpha > \beta$ and $\alpha = \beta$;

- when $K = 0.1$, the defectors reflect the same behavior as above. The ER of collaborators instead is maximum for $f = 1.0$ when $\alpha < \beta$, and for $f = 0.8$ when $\alpha > \beta$ and $\alpha = \beta$.

In scenario C2, shown in Fig. 3.23, we have that:

- when $K = 0.001$, i.e. in a low-trace condition, the ER of defectors is maximum for $f = 0.9$ if $\alpha < \beta$ and if $\alpha > \beta$. If $\alpha = \beta$ is maximum for $f = 0.8$. The ER of collaborators is maximum when $f = 1.0$ only if $\alpha < \beta$. If $\alpha > \beta$ and if $\alpha = \beta$ the maximum is respectively for $f = 0.9$ and $f = 0.8$;
- when $K = 0.1$, the ER of defectors is maximum for $f = 0.9$ but only

Figure 3.23: Percentage of exited agents per type for scenario C2.

if $\alpha < \beta$. If $\alpha > \beta$ and $\alpha = \beta$, the maximum is for $f = 0.7$.The ER of collaborators is maximum when $f = 1.0$ only if $\alpha < \beta$. If $\alpha > \beta$ and if $\alpha = \beta$ the maximum is for $f = 0.7$.

These findings highlight the significance of the interplay between trace value, weight on information, and the composition of agent types within a crowd. By manipulating the trace value and the weight assigned to information, crowd managers and policymakers can potentially influence the exit rate of agents, thereby shaping crowd behavior and achieving desired objectives.

The *paths costs* plots represent, for both types of agents, the cost of their paths, from the start to the endpoint of the environment. All the plots are normalized with respect to the ER.

Beginning with the inspection of the scenario A1, reported in Fig. 3.24,

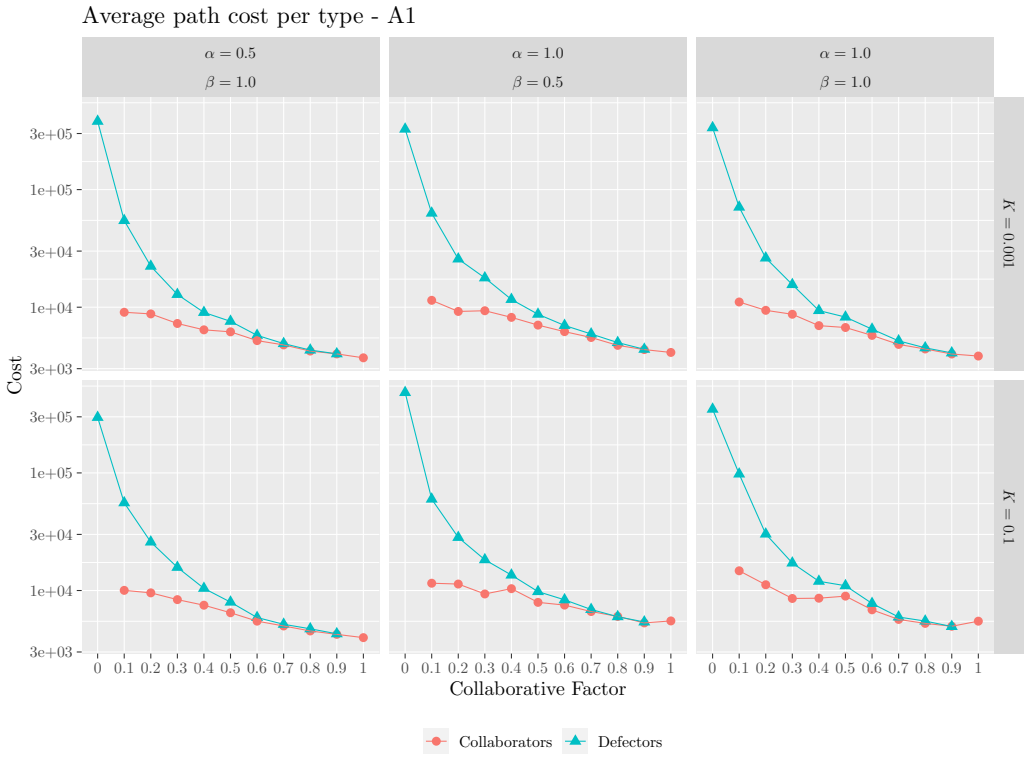


Figure 3.24: Average path cost per agent type for scenario A1.

it is possible to assert that:

- when $K = 0.001$, i.e. low trace, collaborators and defectors find cheaper and cheaper paths for all values of f, independently of the configurations of α and β, and their best tracks are for $f = 1.0$ and $f = 0.9$, respectively;

- when $K = 0.1$, i.e. high trace, only defectors improve their performance for all values of f, independently of the configurations of α and β, and their best paths are for $f = 0.9$. Best paths of the collaborators are instead for $f = 1.0$ when $\alpha < \beta$, and for $f = 0.9$ when $\alpha > \beta$ and $\alpha = \beta$.

In the scenario A2, Fig. 3.25, we may observe that:

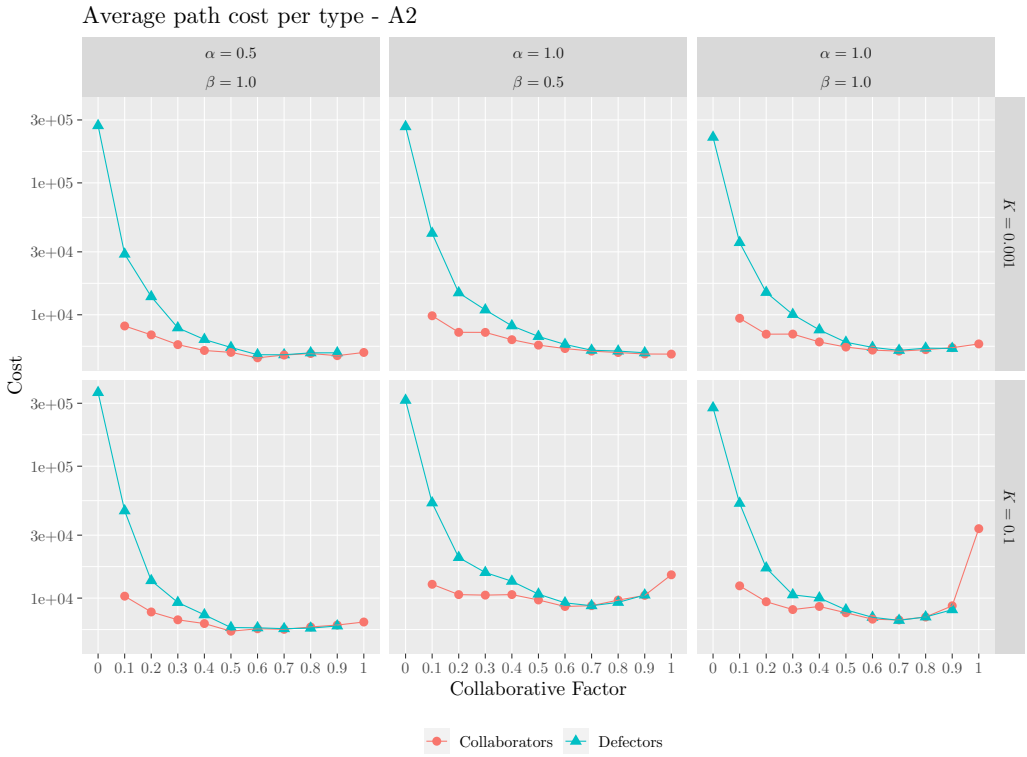


Figure 3.25: Average path cost per agent type for scenario A2.

- when $K = 0.001$, collaborators and defectors find cheaper and cheaper paths for all values of f only for $\alpha > \beta$, and, in particular, their best paths are, respectively, for $f = 1.0$ and $f = 0.9$. When $\alpha < \beta$, both defectors' and collaborators' best paths are obtained at $f = 0.6$, and $f = 0.7$ for $\alpha = \beta$;
- when $K = 0.1$, the best paths are obtained at $f = 0.7$ for both defectors and collaborators, independently of the configurations of α and β. However, it is noted an accentuated worse performance of collaborators for $f = 1.0$, especially when $\alpha = \beta$.

In scenario A1, Fig. 3.24, the path cost of both types of agents increases moving from the top left plot to the bottom right plot. So, moving from a configuration with $K = 0.001$ and $\alpha < \beta$ to a configuration with $K = 0.1$ and $\alpha = \beta$. We recall that the agents aim to minimize their paths' costs.

As one has seen for the absolute and relative count plots, for $K = 0.001$, i.e., when the trace value is low, the path cost of both types of agents is more or less the same regardless of whether, during their decision-making process, they give more weight to the discovered information η ($\alpha < \beta$), to the trace ($\alpha > \beta$) or equal weight to both parameters ($\alpha = \beta$). In particular, collaborators find their best paths when $f = 1.0$, while defectors when $f = 0.9$. This behavior is more pronounced when the trace value is higher, i.e. when $K = 0.1$ and it can be seen that the path costs of both types of agents increase moving from the plot on the left to the one on the right. The path costs of the agents are lower if, during their decision-making process, they give more weight to the discovered information η ($\alpha < \beta$) than the same weight to information and trace ($\alpha = \beta$). Collaborators' paths' costs go from a minimum reached for $f = 1.0$ when $\alpha < \beta$ to a minimum reached for $f = 0.9$ when $\alpha > \beta$ and $\alpha = \beta$. Also for the path costs, we can say that the presence of defectors is useful only under certain conditions, i.e. when the trace released is high and, at the same time, the agents give equal weight to the trace and to the information because, in these conditions, the path cost of collaborators is lower than when the crowd is composed entirely of collaborative agents. In scenario A2 (Fig. 3.25) the behavior of the agents is very similar to that discussed for A1, with slightly worse performance than the latter, especially when $K = 0.1$ and $\alpha = \beta$. In this case, is very clear to note the positive influence of defectors agents since collaborators' path costs are better when $f = 0.9$, i.e. when the crowd is mainly but not totally collaborative. In general, also here, the performance of the agents worsens when moving from the upper left plot to the lower right plot, that is when one passes from a condition of low trace and greater weight to information to a condition of high trace and equal weight to trace and information.

In scenario B1, Fig. 3.26, we have that:

- when $K = 0.001$, only defectors find cheaper and cheaper paths for all values of f, independently of the configurations of α and β, and their best paths are respectively for $f = 0.9$. Collaborators instead improve their performance for all values of f only when $\alpha < \beta$ and $\alpha > \beta$. In these cases, their best path is for $f = 1.0$. In the last case,

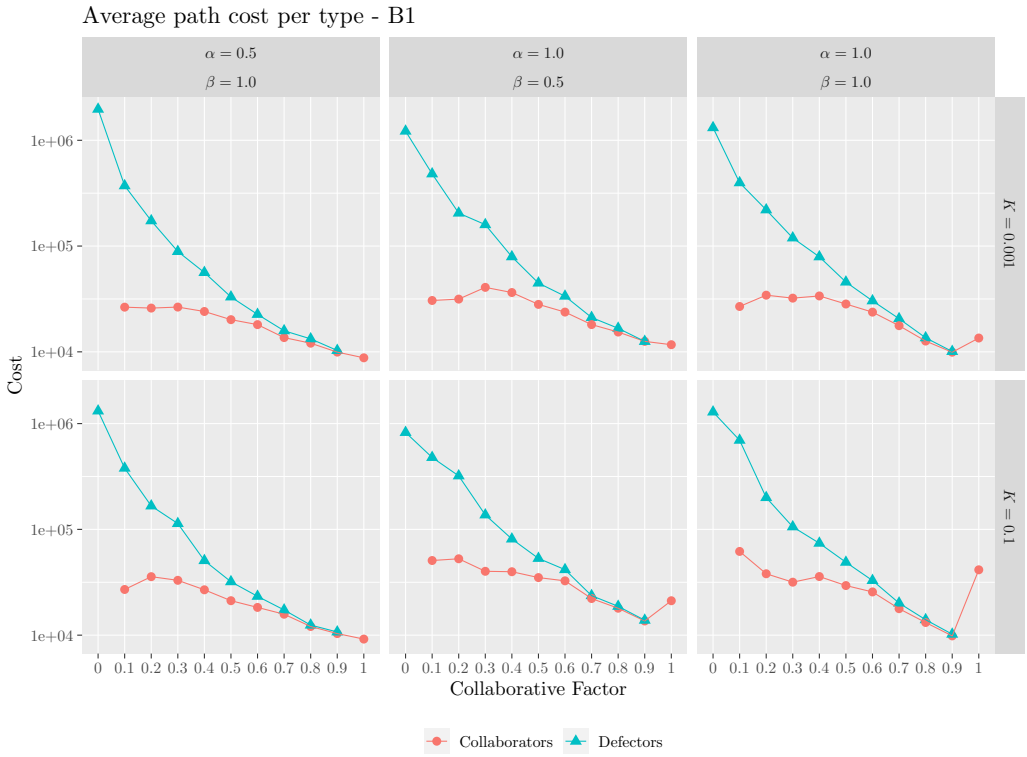


Figure 3.26: Average path cost per agent type for scenario B1.

when $\alpha = \beta$, their best path is for $f = 0.9$;

- when $K = 0.1$, also in this case, only defectors improve their performance for all values of f, independently of the configurations of α and β, and their best paths are for $f = 0.9$. Collaborators' best paths are for $f = 1.0$ when $\alpha < \beta$, for $f = 0.9$ when $\alpha > \beta$ and $\alpha = \beta$. Also here is noted an accentuated worse performance of collaborators for $f = 1.0$.

For the scenario B2, Fig. 3.27, also in this case emerges that:

- when $K = 0.001$, only defectors find cheaper and cheaper paths for all values of f, independently of the configurations of α and β and their best paths are for $f = 0.9$. Collaborators' best paths are for $f = 1.0$ when $\alpha < \beta$ and $\alpha > \beta$, and for $f = 0.6$ when $\alpha = \beta$;

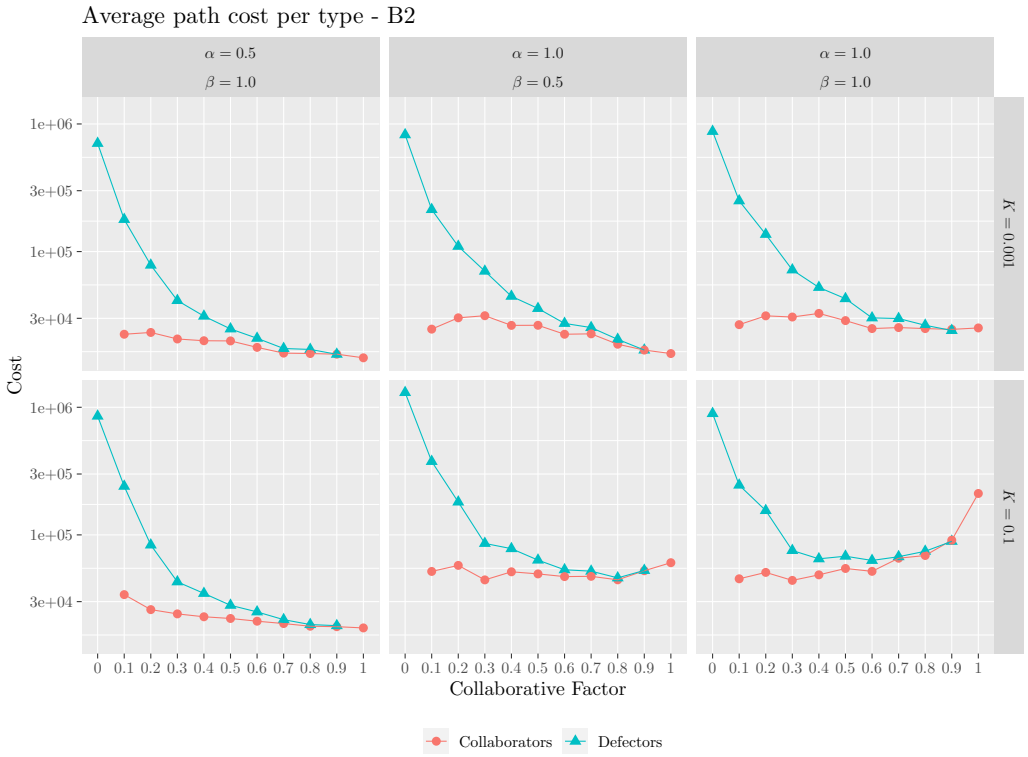


Figure 3.27: Average path cost per agent type for scenario B2.

- when $K = 0.1$, both defectors and collaborators find cheaper and cheaper paths for all values of f only when $\alpha < \beta$ and their best paths are respectively for $f = 0.9$ and $f = 1.0$. If $\alpha > \beta$, collaborators' best paths are for $f = 0.3$, defectors' ones for $f = 0.8$. Finally, when $\alpha = \beta$, collaborators' best paths are for $f = 0.1$ and defectors ones for $f = 0.6$.

Inspecting scenario C1, Fig. 3.28, we have that:

- when $K = 0.001$, i.e. when the trace is low, only defectors find cheaper and cheaper paths for all values of f, independently of the configurations of α and β, and, specifically, their best paths are respectively for $f = 0.9$. Collaborators' best paths are for $f = 1.0$ if $\alpha < \beta$, and for $f = 0.9$ if $\alpha > \beta$ and $\alpha = \beta$;

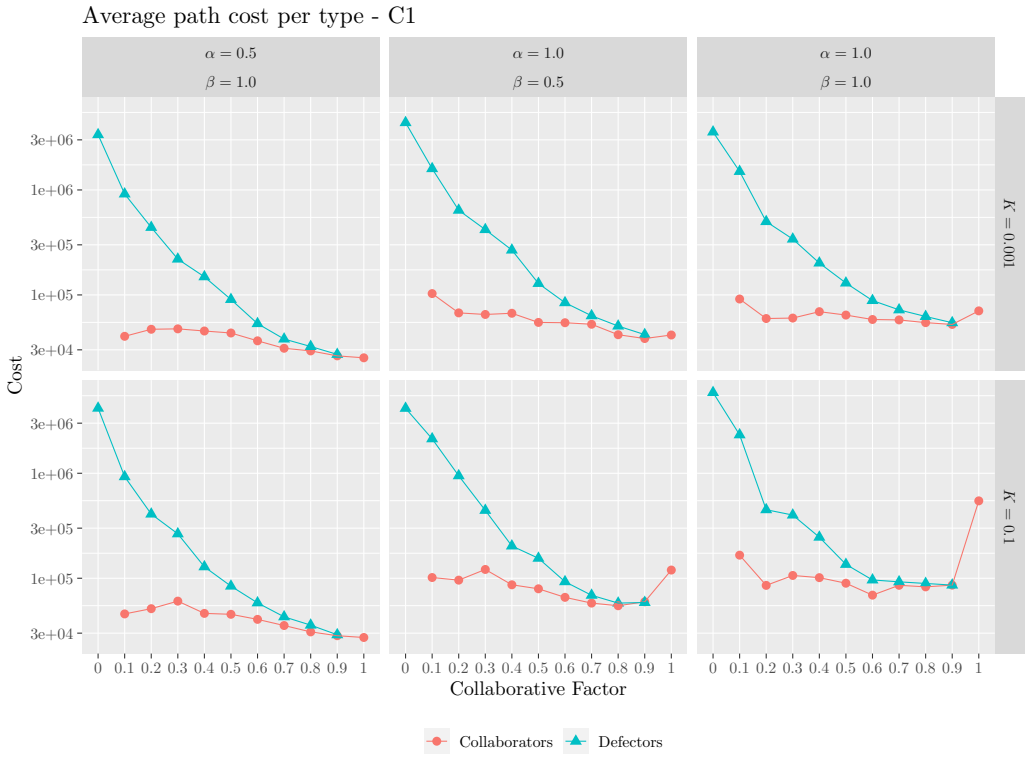


Figure 3.28: Average path cost per agent type for scenario C1.

- when $K = 0.1$, also here, only defectors improve their performance for all values of f regardless of the configurations of α and β, and their best paths are for $f = 0.9$. Collaborators' best paths are for $f = 1.0$ when $\alpha < \beta$, for $f = 0.9$ when $\alpha > \beta$ and for $f = 0.6$ when $\alpha = \beta$. Here is noted an accentuated worse performance of collaborators for $f = 1.0$.

Finally, analyzing the outcomes of scenario C2, shown in Fig. 3.29, it is possible to assert also in this scenario that:

- when $K = 0.001$, only defectors find cheaper and cheaper paths for all values of f, independently of the configurations of α and β and their best paths are for $f = 0.9$ if $\alpha < \beta$ and $\alpha > \beta$. In this last case, however, we mention a small worse in their performance for $f = 0.8$. When $\alpha = \beta$, their best paths are for $f = 0.8$. Collaborators'

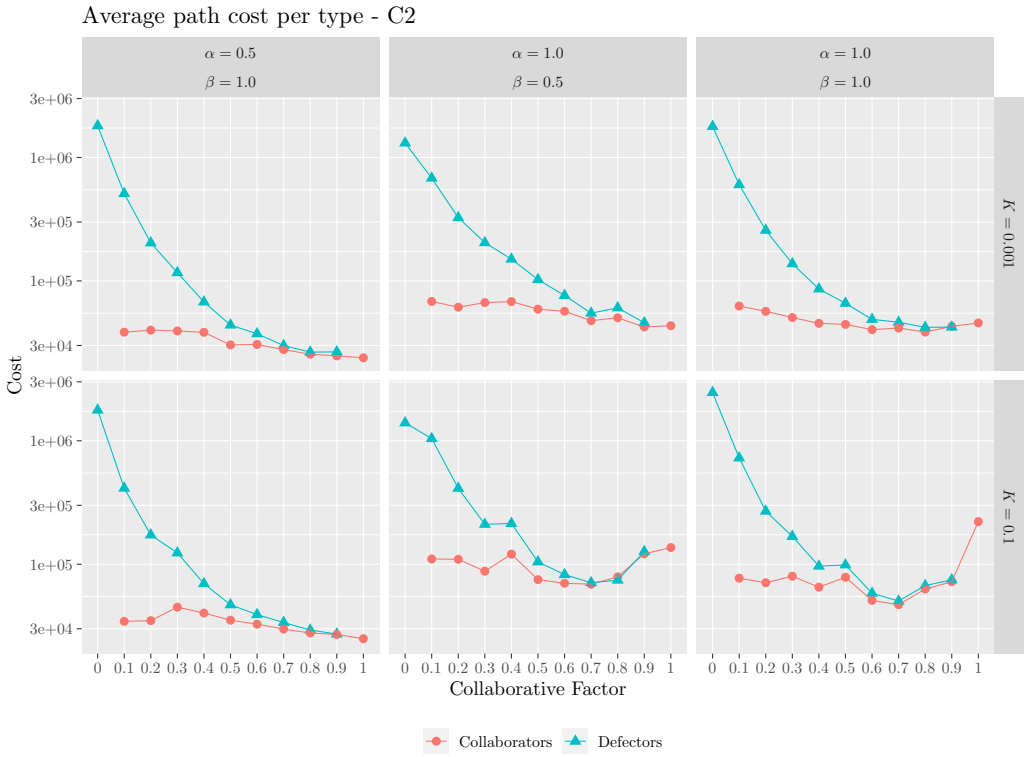


Figure 3.29: Average path cost per agent type for scenario C2.

best paths are for $f = 1.0$ when $\alpha < \beta$, for $f = 0.9$ if $\alpha > \beta$, and for $f = 0.8$ when $\alpha = \beta$;

- when $K = 0.1$, both defectors and collaborators find cheaper and cheaper paths for all values of f only when $\alpha < \beta$ and their best paths are respectively for $f = 0.9$ and $f = 1.0$. Also, in this case, we mention a small worse in collaborators' performance for $f = 0.3$. If $\alpha > \beta$, both collaborators' and defectors' best paths are for $f = 0.7$. We have the same findings if $\alpha = \beta$.

In real-life scenarios, the path cost may represent the effort or resources expended by agents to navigate through the crowd and reach their destinations. Minimizing path costs is a desirable objective for both crowd managers and individuals within the crowd. The presence of defectors can have a positive influence on collaborative agents by poten-

tially improving their path costs, particularly when the trace value is high and the weight on information is balanced. This phenomenon can be explained by considering the role of defectors in diversifying the decision-making process within the crowd. Understanding the implications of these findings is crucial for crowd management and decision-making in real-life scenarios because they can leverage this knowledge not only to optimize crowd behavior and decision-making, but also to design crowd management strategies in different scenarios such as large events, transportation systems, or emergency evacuations.

The *exit times* represent how fast both types of agents have reached the exit. All reported plots are normalized with respect to the ER, already defined before.

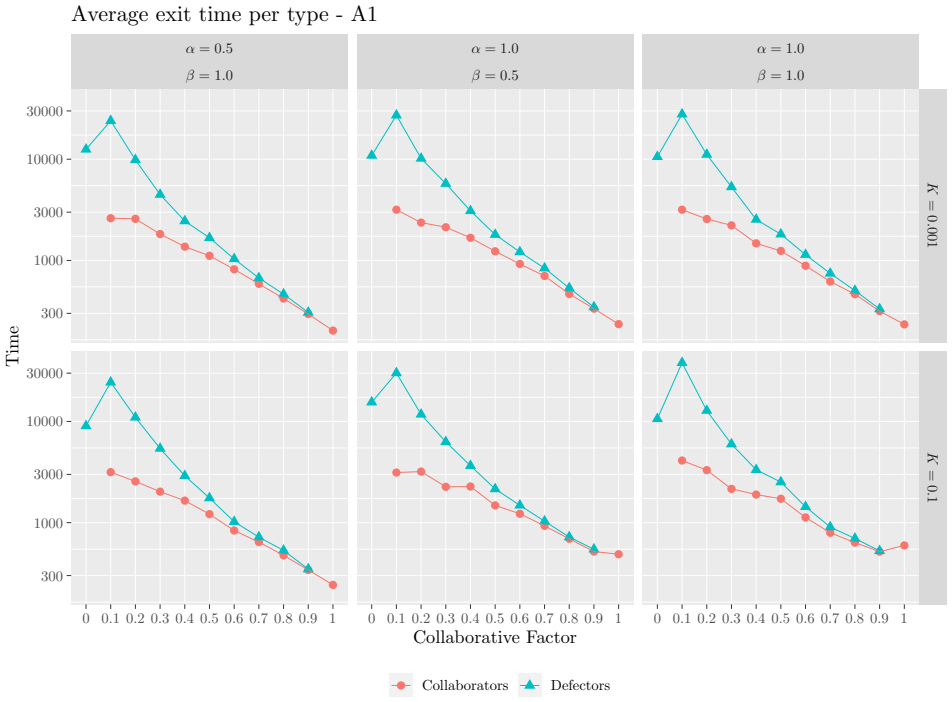


Figure 3.30: Average exit time per agent type for scenario A1.

Inspecting scenario A1, Fig. 3.30, we have that:

- when $K = 0.001$, collaborators and defectors exit faster and faster for all values of f, independently of the configurations of α and β,

and their best performances are respectively for $f = 1.0$ and for $f = 0.9$;

- when $K = 0.1$, only defectors exit faster and faster for all values of f, independently of the configurations of α and β, and their best performances are for $f = 0.9$. Collaborators, however, improve their performance only if $\alpha < \beta$ and their best performance is for $f = 1.0$. If $\alpha > \beta$ and $\alpha = \beta$ they exit faster for $f = 0.9$ and worse their performance for $f = 1.0$.

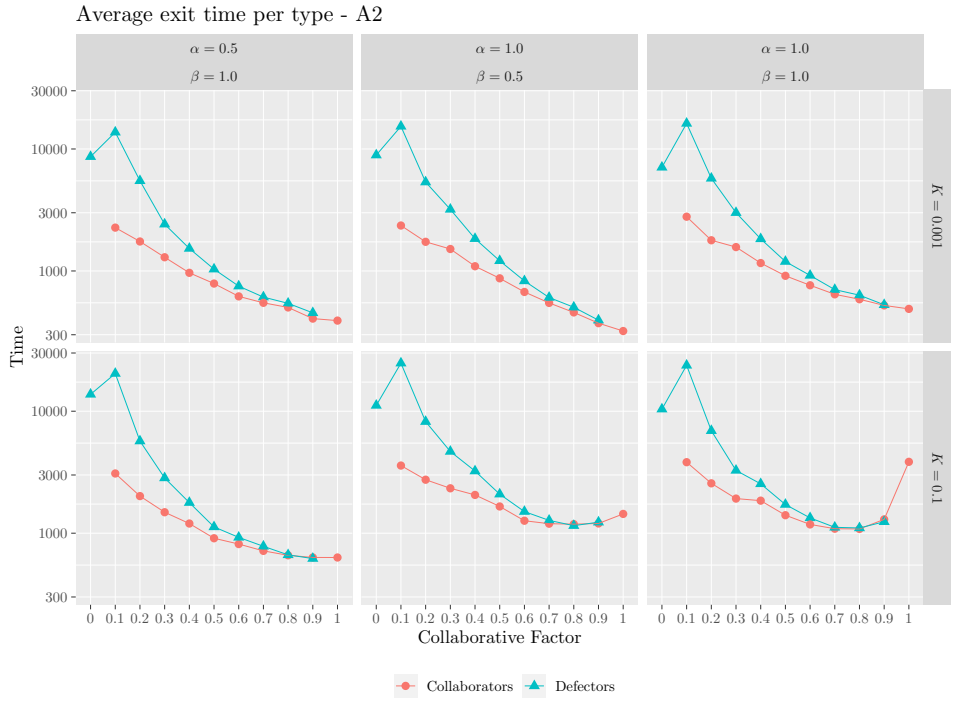


Figure 3.31: Average exit time per agent type for scenario A2.

For the scenario A2, displayed in Fig. 3.31, we have that:

- when $K = 0.001$, collaborators and defectors exit faster and faster for all values of f only if $\alpha > \beta$, and their best performances are respectively for $f = 1.0$ and for $f = 0.9$. If $\alpha < \beta$ their best performances are both for $f = 0.6$, whilst for $\alpha = \beta$, are both at $f = 0.7$;

- when $K = 0.1$, defectors' and cooperators' best performance are at $f = 0.7$, independently of the configurations of α and β. It is noted a worse in collaborators' performance especially when $f = 1.0$ and $\alpha = \beta$.

In the overall, in scenario A1 (Fig. 3.30) as for the path cost, the agents aim to minimize the exit time and the one of both types, increases moving from the top left plot to the bottom right plot. So, moving from a configuration with $K = 0.001$ and $\alpha < \beta$ to a configuration with $K = 0.1$ and $\alpha = \beta$. As one has seen for the absolute and relative count plots, and for the path cost, for $K = 0.001$, i.e., when the trace value is low, the exit time of both types of agents is more or less the same regardless of whether, during their decision-making process, they give more weight to the discovered information η ($\alpha < \beta$), to the trace ($\alpha > \beta$) or equal weight to both parameters ($\alpha = \beta$). In particular, collaborators exit faster when $f = 1.0$, while defectors do when $f = 0.9$. This behavior is more pronounced when the trace value is higher, i.e. when $K = 0.1$ and it can be seen that the exit time of both types of agents increases moving from the plot on the left to the one on the right. The exit times of the agents are lower if, during their decision-making process, they give more weight to the discovered information η ($\alpha < \beta$) than the same weight to information and trace ($\alpha = \beta$). Collaborators' exit times go from a minimum reached for $f = 1.0$ when $\alpha < \beta$ to a minimum reached for $f = 0.9$ when $\alpha = \beta$. Also for this metric, we can say that the presence of defectors is useful only under certain conditions, i.e. when the trace released is high and, at the same time, the agents give equal weight to the trace and to the information because, in these conditions, the exit times of collaborators are lower than when the crowd is composed entirely of collaborative agents. In scenario A2, in Fig. 3.31, the behavior of the agents is very similar to that discussed for A1, with slightly worse performance than the latter, especially when $K = 0.1$ and $\alpha = \beta$. In this case, as we have seen for the path cost, is very clear to note the positive influence of defectors agents since collaborators' exit times are better when $f = 0.9$, i.e. when the crowd is mainly but not totally collaborative. In general, also here, the performance of the agents worsens when moving from the upper left plot to the lower right plot, that is when one passes from a condition of

low trace and greater weight to information to a condition of high trace and equal weight to the trace and information.

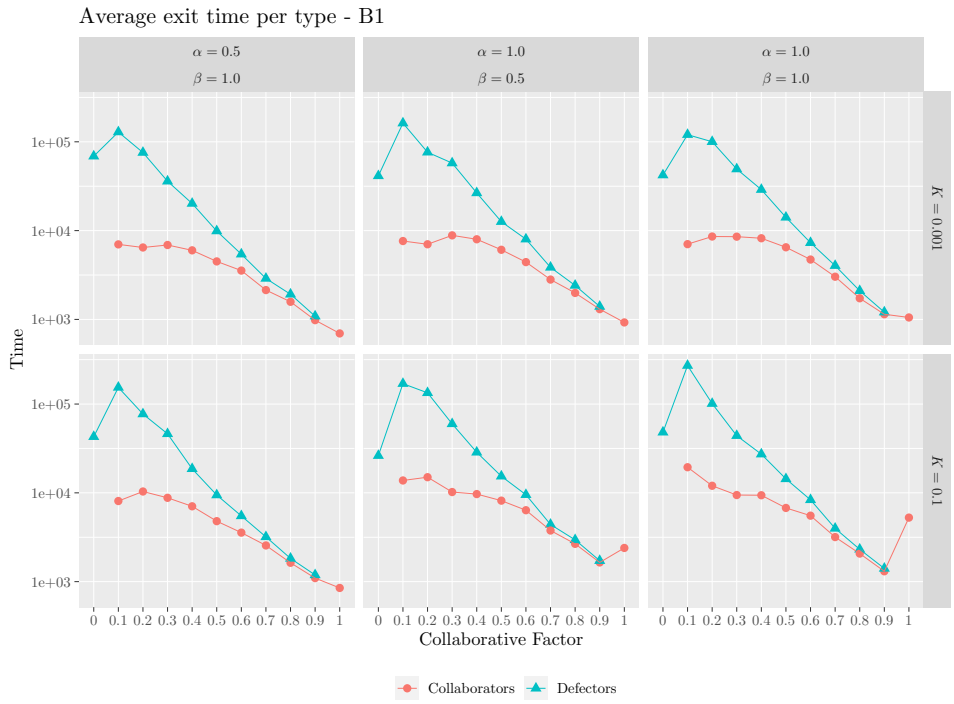


Figure 3.32: Average exit time per agent type for scenario B1.

From the analysis of the scenario B1, Fig. 3.32, we have that:

- when $K = 0.001$, i.e. when the trace is low, defectors exit faster and faster for all values of f, independently of the configurations of α and β, and their best exit times are for $f = 0.9$. Collaborators, improve their exit times for all values of f but only if $\alpha < \beta$ and $\alpha > \beta$. In these cases, their best exit times are for $f = 1.0$. If $\alpha = \beta$ they exit faster for $f = 0.9$;
- when $K = 0.1$ we have the same results as above. It is worth pointing out also here, a worse in collaborators' performance especially when $f = 1.0$ and $\alpha = \beta$.

For scenario B2, Fig. 3.33, we observe that:

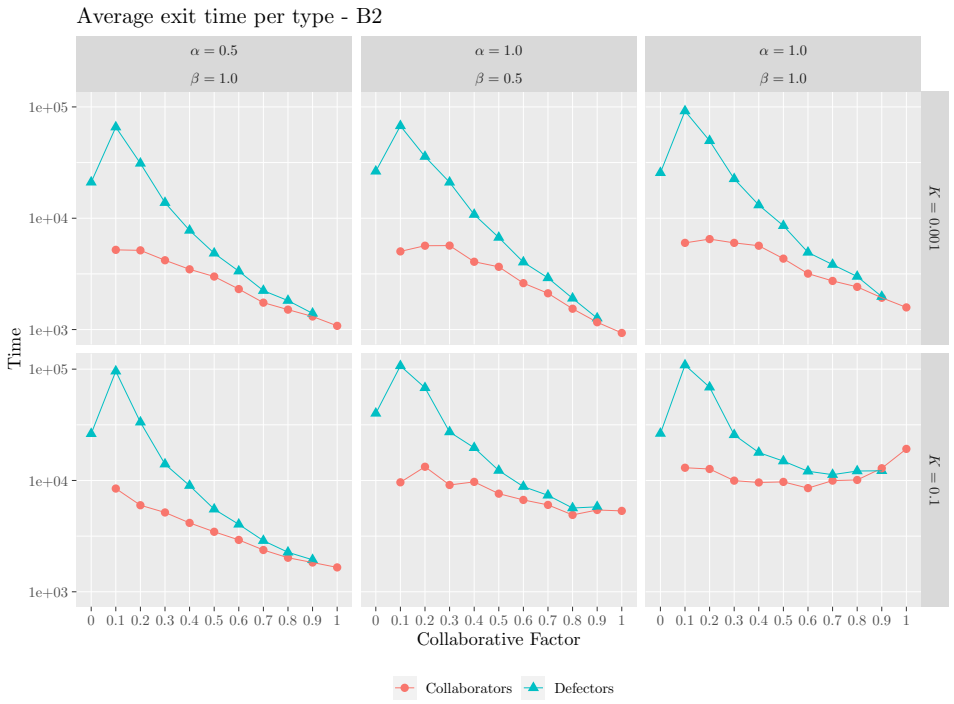


Figure 3.33: Average exit time per agent type for scenario B2.

- when $K = 0.001$, also in this scenario, defectors exit faster and faster for all values of f, independently of the configurations of α and β, and their best exit times are for $f = 0.9$. Collaborators improve their exit times for all values of f but only if $\alpha < \beta$ and $\alpha > \beta$; in these cases, their best exit times are for $f = 1.0$. If $\alpha = \beta$ instead they exit faster for $f = 0.6$;

- when $K = 0.1$, collaborators and defectors exit faster and faster for all values of f only if $\alpha < \beta$ and their best times are respectively for $f = 1.0$ and $f = 0.9$. If $\alpha > \beta$, collaborators are faster for $f = 0.3$, defectors for $f = 0.8$. If $\alpha = \beta$, collaborators are faster for $f = 0.1$ while defectors for $f = 0.6$.

In scenario C1, whose outcomes are shown in Fig. 3.34, it is possible to observe, also in this case, that:

- when $K = 0.001$, defectors exit faster and faster for all values of f,

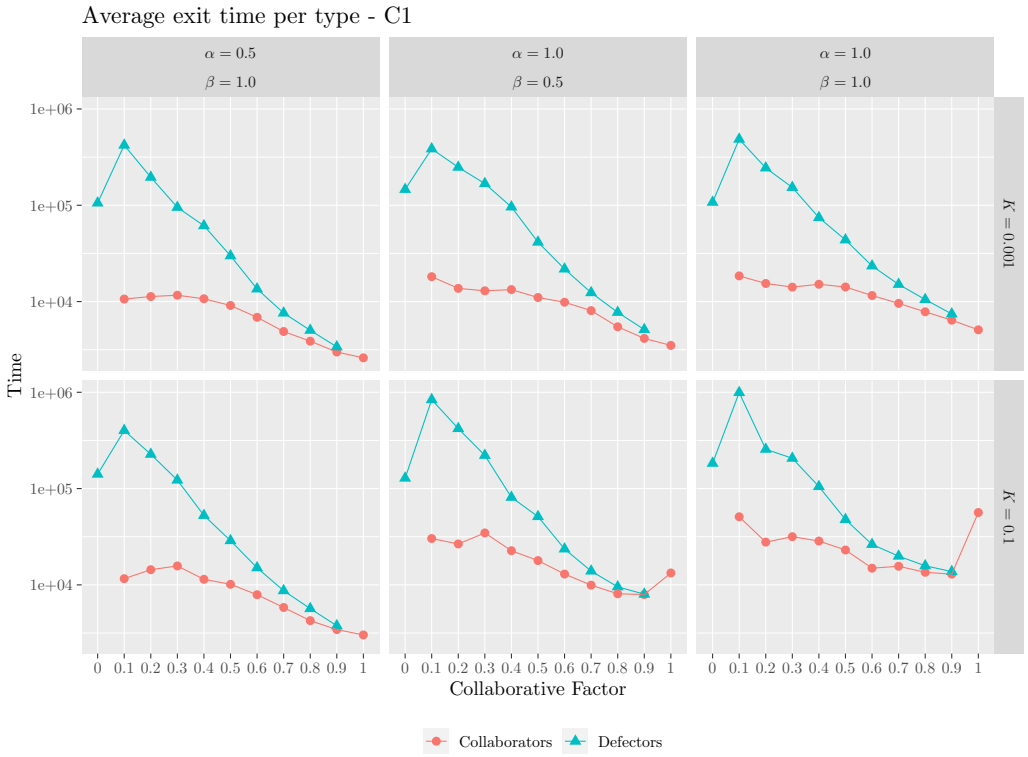


Figure 3.34: Average exit time per agent type for scenario C1.

independently of the configurations of α and β, and their best exit times are for $f = 0.9$. Collaborators improve their exit times for all values of f but only when $\alpha < \beta$. In this case, the best exit time is for $f = 1.0$. If $\alpha > \beta$ and $\alpha = \beta$ they exit faster for $f = 0.9$;

- when $K = 0.1$ we have more or less the same results as above. Defectors exit faster and faster for all values of f, independently of the configurations of α and β, and their best exit times are for $f = 0.9$. It is noted, also here, a worse in collaborators' performance especially when $f = 1.0$. In particular, when $\alpha > \beta$ their exit times are the best for $f = 0.8$. When $\alpha = \beta$ the best exit times are for $f = 0.6$

Investigating the last scenario C2, Fig. 3.35, also in this case we have that:

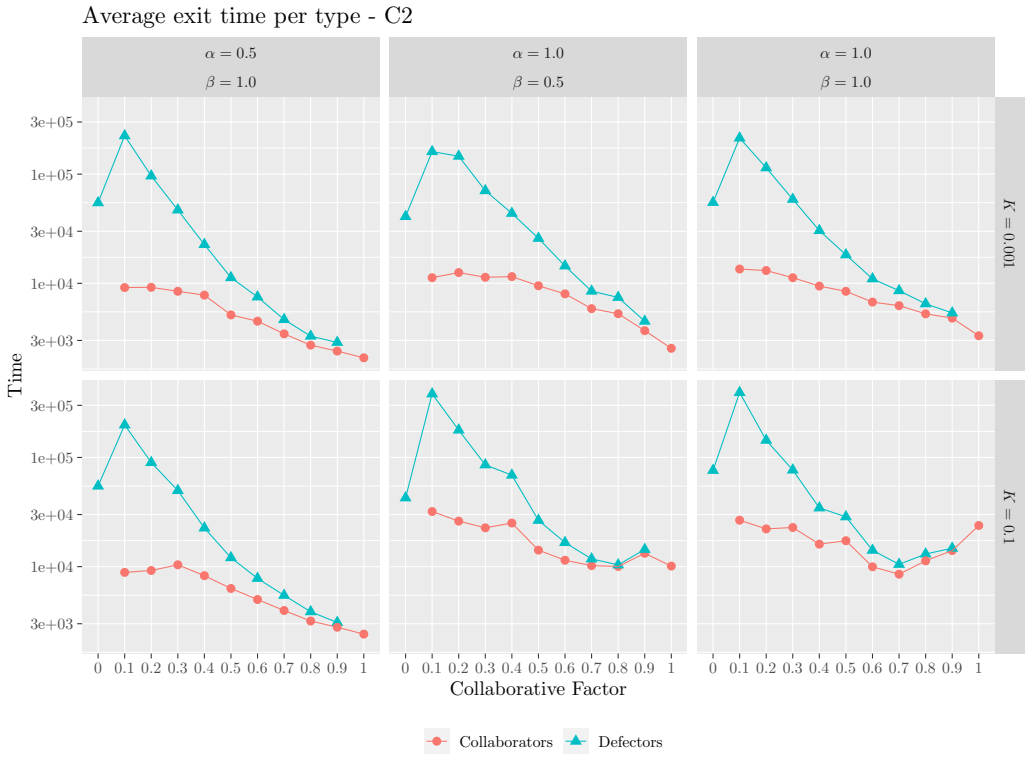


Figure 3.35: Average exit time per agent type for scenario C2.

- when $K = 0.001$, defectors exit faster and faster for all values of f, independently of the configurations of α and β, and their best exit times are for $f = 0.9$. It is noted, here, a small worse in performances for $f = 0.8$. Collaborators improve their exit times for all values of f but only if $\alpha < \beta$ and $\alpha > \beta$. In these cases, their best exit times are for $f = 1.0$. If $\alpha = \beta$ they exit faster for $f = 0.8$;
- when $K = 0.1$, collaborators and defectors exit faster and faster for all values of f only for $\alpha < \beta$ and their best times, are respectively for $f = 1.0$ and $f = 0.9$. If $\alpha > \beta$, collaborators are faster for $f = 0.7$ whilst defectors for $f = 0.8$. Finally, when $\alpha = \beta$, both collaborators and defectors are faster for $f = 0.7$.

Exit times play a critical role in optimizing crowd flow and ensuring smooth and efficient movement. Our findings suggest that the presence

of defectors in crowd scenarios plays a significant role in shaping the outcomes, particularly in terms of exit times. In particular, they indicate that under certain conditions, such as high trace value and equal weight assigned to trace and information, the presence of defectors can lead to lower exit times for collaborative agents. This highlights the importance of the interplay between trace value, weight on information, exit times, and composition of the crowd. The utility of defectors is context-dependent and is most pronounced when there is abundant information available and when agents give equal weight to trace and information.

3.3.2 Conclusions

In conclusion, the studies conducted with the second version of the model have contributed to answering the established research question: What is the interplay between collaboration and competition in social complex systems? Various kinds of analyses were performed, each focusing on a specific aspect of the problem.

In detail, the type analysis reveals that the crowd's performance improves as the collaborative factor but the optimal f value for peak performance varies across scenarios. In general, fully collaborative crowds ($f = 1.0$) and fully defector crowds ($f = 0.0$) result in the worst performance, as expected. The percentage of agents successfully reaching the exit improves with increasing f, but the trends vary based on agent type and scenario. Collaborators outperform defectors when in the minority, but defectors leverage collaborator information when outnumbered. Path costs reveal varying trends with increasing f. Defectors generally improve their path costs with higher f, reaching a minimum at optimal f values in all scenarios. Collaborators' path costs vary, reaching a minimum at different f values in each scenario, with an increase at $f = 1.0$. Exit times show intricate patterns with changing f. Both collaborators and defectors improve exit times with increasing f until reaching optimal values in all scenarios. An unexpected increase in exit times occurs at $f = 1.0$, indicating suboptimal performance for fully collaborative crowds. From the Group analysis it appears that the exit time decreases with respect to the group number, indicating that later-evacuating groups somehow benefit from information left by earlier groups. Gener-

ally, group 1, for instance, demonstrates better performance at higher f values, suggesting that agents in this group exploit path information more effectively, especially when the majority of the crowd is collaborative. Group 10, instead, consistently improves its performance with increasing f, except for $f = 1.0$. This implies that the last group can utilize path information better when the crowd is predominantly, but not entirely, collaborative. Similar conclusions are drawn from the path cost as it decreases with higher collaboration and group number, except for $f = 1.0$, where absolute collaboration seems less effective. The heat maps indicate the number of agents reaching the exit per unit time.
Interestingly, the highest performance is not observed when all agents are collaborative but when there's a mix, including defectors. This quantity also increases concerning the group number, highlighting the beneficial influence of earlier-evacuating groups. The overall analysis revealed that a fully collaborative crowd tends to demonstrate suboptimal performance. The complexity of crowd dynamics becomes evident when a combination of collaborators and defectors is present. Interestingly, the disruptive actions introduced by defectors paradoxically enhance the overall efficiency of the exit process. While collaborative efforts remain crucial, the disruptive actions of defectors play an essential role in refining the crowd's navigation strategy. In this sense, the destructive actions of the defectors act as a pruning mechanism, eliminating less favorable paths and steering the crowd towards more efficient routes. Consequently, the crowd, despite incorporating non-collaborative elements, collectively attains superior outcomes in exit time, path cost, and the number of agents reaching the exit point. This interplay between collaborative and non-collaborative behavior creates a dynamic and adaptable crowd, showcasing the ability to optimize performance across different conditions. From the sensitivity analysis, it can be observed that the previous analyses continue to be valid but only under specific circumstances marked by $K = 0.1$ and $\alpha = \beta$, indicating a scenario where both the trace (τ_{ij}) and the information (η_{ij}) carry equal weight, coupled with a high trace value. In such instances, both collaborators and defectors exhibit suboptimal performance, illuminating the impact of balancing trace and information in the decision-making process. However, the positive influence of defectors on collaborators is notable only within this specific configuration. This

implies that the presence of defectors can be advantageous for collaborators under circumstances where both trace and information are given equal importance. In contrast, when the trace is at a low value ($K = 0.001$) or agents assign greater importance to information ($\alpha < \beta$), collaborators consistently outperform defectors. This highlights the intricate interplay between trace and information, emphasizing that the performance of both collaborators and defectors is intricately linked to the specific parameter values of α and β that govern their decision-making processes. In scenarios where the trace is low or information is prioritized, collaborators demonstrate superior adaptability and effectiveness in navigating the environment compared to defectors. In essence, the interplay between collaboration and competition in social complex systems is revealed to be intricate and nuanced in the studies conducted. Collaboration emerges as a pivotal factor driving crowd performance, leading to enhanced outcomes overall. However, maintaining the optimal balance between collaboration and competition is crucial, as extremes of either fully collaborative or defector-dominated crowds yield suboptimal results. Notably, collaborators tend to outperform when they constitute a minority within the crowd, while defectors strategically exploit collaboration when they form the majority. Furthermore, the analysis uncovers also the complex dynamics of information transfer within crowd behavior. Later-evacuating groups benefit significantly from the information left behind by earlier-evacuating groups, underscoring the importance of collective knowledge sharing. Additionally, the sensitivity analysis highlights the intricate interplay between trace values and information. Collaborators demonstrate heightened effectiveness when trace values are low or when they weigh more information in decision-making processes.

In conclusion, the research underscores that collaboration plays a central role in shaping crowd behavior and performance in social complex systems. However, its effectiveness relies on maintaining a balanced and adaptive approach, considering the interactions between collaborators and defectors and the subtle dynamics of trace and information. In other words, collaboration drives the dynamic through information sharing, while competition (defection) enhances the dynamic by simplifying the environment.

3.4 The third crowd model

In the third model, which will be presented in an upcoming publication [22], the research question was broadened to address the following: What impact does group size have on the behavior of collaborative and defector agents within the overall crowd? Observations of group formation in pedestrian crowds highlight the considerable influence that groups exert on crowd dynamics [46], including their effect on evacuation times [89]. Additionally, the mechanism for agent exploration of the environment underwent refinement to incorporate the agent's capability to estimate the weight of an edge when it is not directly visible, as illustrated in Fig. 3.36. Supposed the green circle to be the piece of information present on the endpoint of an edge (i, j). An agent k placed on the node i sees the information only if it is present on the nearest endpoint, otherwise, it doesn't see it. This modification aimed to make the behavior of agents more closely resemble human-like reasoning. To do this, adjustments were made to the desirability $\eta_{ij}(t)$ in Eq. 1.1, which determines how much an edge (i, j) is promising at a given time t.

In detail, as it is established in the second crowd model, agents employ a navigation strategy that combines the use of trace and the calculated desirability of edges to make informed decisions about where to go. Traces act as a communication device among agents, directing them towards paths with higher concentrations of traces. Conversely, desirability is internally determined based on acquired information $I_{ij}(t)$ about edge weights, achieved through two mechanisms.

The first memory mechanism employed by the agents involves traversing edges and storing the weights in their memory as *prior knowledge* $\bar{w}_p^k$. As agents explore the environment, they accumulate information about the weights of the edges they traverse. The *prior knowledge* represents the average weight of the visited edges, computed by summing the weights along a generic path ($\pi(t)$) and dividing by the number of edges in that path (n). This process, outlined in Equation 3.4, enables the agents to establish a historical understanding of the environment. The *prior knowledge* mechanism allows the agents to form preferences for certain paths based on their past experiences. In essence, this memory mechanism reflects the agents' ability to internalize and use information about previously

encountered paths, shaping their decision-making in navigating the environment.

$$\bar{w}_p^k = \frac{1}{n} \sum_{i=1}^{n} w(\pi_i, \pi_{i+1}); \tag{3.4}$$

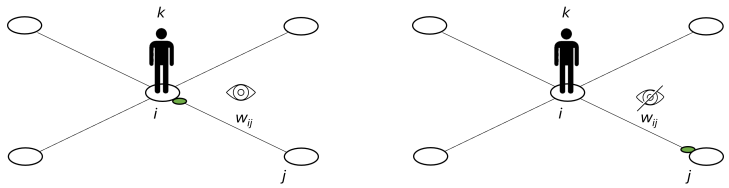
The second memory mechanism involves the acquisition of information about edge weights at the nearest endpoint, leading to the formation of their *local knowledge* $\bar{w}_l^k$. Agents discern information ($I_{ij}(t)$) about the weights of edges in their immediate vicinity, considering the number of neighboring edges (m) at the position where this information is present. The *local knowledge* is then determined as the average weight of these neighboring edges, as expressed in Equation 3.5. This mechanism allows agents to adapt their decision-making based on real-time information acquired from their immediate surroundings. By considering the specific context at the nearest endpoint, agents enhance their awareness of the current environment and adjust their desirability calculations accordingly. The combination of *prior knowledge* and *local knowledge* mechanisms enables agents to strike a balance between past experience and current information, facilitating efficient navigation and adaptation to their dynamic surroundings.

$$\bar{w}_l^k = \frac{1}{m} \sum_{i=1}^{m} w(\pi_i, \pi_{i+1}), \tag{3.5}$$

The overall desirability value $\eta_{ij}(t)$ is then determined using a decision tree (Equation 3.6), considering the availability of information $I_{ij}(t)$ and the type of knowledge.

$$\eta_{ij}(t) = \begin{cases} \frac{1}{w_{ij}} & \text{if } I_{ij} \neq 0 \text{ and } T \neq 0 \\ \frac{1}{\bar{w}} & \text{if } I_{ij} = 0 \text{ and } T \neq 0 \\ 1 & \text{if } I_{ij} = 0 \text{ and } T = 0 \end{cases} \tag{3.6}$$

The *global knowledge* $\bar{w}$ is the mean of prior and local knowledge, as shown in Equation 3.7. When an agent is on a node i, it calculates $\eta_{ij}(t)$ considering the edge weight, with higher desirability for lower weights.



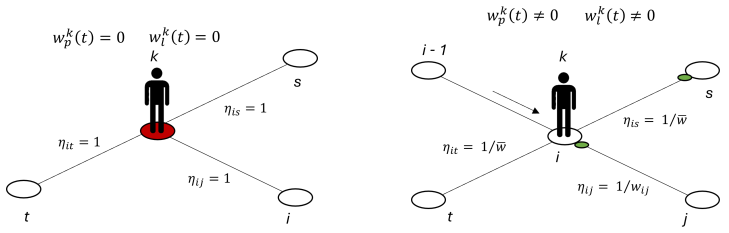
(a) (b)

Figure 3.36: Agent k positioned on node i perceives information $I_{ij}(t)$ (green dot) regarding the weight of edge (i, s) only when it is located at the nearest endpoint (Figure 3.36a), otherwise, it remains unaware of it (Figure 3.36b).

If information $I_{ij}(t)$ is available at either nearest endpoint, the agent inversely relates desirability to the edge weight. In scenarios where information $I_{ij}(t)$ about neighboring edges is absent, the agent estimates desirability as the inverse of its global knowledge. This estimation is derived by averaging prior and local knowledge. Figure 3.37b illustrates this.

$$\bar{w} = \frac{\bar{w}_p^k + \bar{w}_l^k}{2}. \tag{3.7}$$

If information about the weight of neighboring edges (i, j) is missing or not visible but at least one piece of information is available, as represented in Fig 3.37b, the agent will estimate the desirability $\eta_{ij}(t)$ as the inverse of its global knowledge $\bar{w}$ about the environment. This estimation is obtained by averaging the agent's prior and local knowledge, as described in eq. 3.7. However, it's possible for an agent to have no local knowledge $\bar{w}_l^k$ when no information is available about the weight of the neighboring edges. In such a scenario, the agent will evaluate the weight of each edge as the inverse of its prior knowledge $\bar{w}_p^k$. In cases of no prior and local knowledge, as depicted in Figure 3.37a, the agent assigns equal desirability value 1 to all edges. Apart from these modifications, the model largely maintains continuity with the second crowd model, with minor differences in the code. It still features the same two types of agents: collaborators and defectors, each exhibiting their respective behavioral actions. The collaborative factor f remains instrumental, dic-



(a) (b)

Figure 3.37: In Figure 3.37a, an agent k situated at node i assesses the desirability of adjacent edges as equal to 1 in absence of both prior ($w_p^k = 0$) and local knowledge ($w_l^k = 0$). In Figure 3.37b, when agent k reaches node i from node $i-1$, it evaluates adjacent edges. If information $I_{ij}(t)$ on edge weights is either missing or not visible, the desirability is determined as the inverse of global knowledge. When the information $I_{ij}(t)$ is visible, the desirability is assessed as the inverse of the weight of the link.

tating the proportion of collaborative agents relative to defectors within the crowd.

3.4.1 Experiments and results

For the analysis, a graph with $|V| = 225$ and $|E| = 501$ was utilized, where each node connects to up to eight neighbors, and edge weights are real numbers selected uniformly from the range $]0, 1]$. Simulations involved $N = 1000$ agents distributed into Γ groups, with Γ varying from the set $1, 2, 4, 5, 10, 100$. The collaborative factor, denoted by $f \in [0, 1]$, determined the ratio of collaborative to defector agents in a group. Groups could consist of one agent type or a mix, with f representing the fraction of collaborative agents and $(1 - f)$ the fraction of defector agents. Groups started exploration after a fixed time $T_e = |V|$ since the preceding group, and a time limit T_{max} was set for the entire population to reach the exit. This limit is defined as $T_{max} = c \times \Gamma \times |V|$, with c set to 5. Agents had individual time windows to reach the exit, determined by $T_i = T_{max} - (T_e \times (i - 1))$. Time impact was incorporated by gradual trace reduction with a fixed degradation interval of $T_d = |V|$. The updating

rule occurred every T_d ticks with an evaporation rate of $\rho = 0.001$. The initial trace value was $\tau_{ij}(t = 0) = 1.0$. Destruction and repair probabilities were equal for both agent types. To assess the impact of group sizes, experiments were conducted, varying the collaborative factor f from 0.0 to 1.0 in increments of 0.1, with 10 independent simulations for each f value and all preset Γ values.

Two distinct analyses were conducted: an **overall analysis**, where the system's performance was compared across groups of different sizes, and a **type analysis**, where the performance of collaborators and defectors within the same groups was compared. Both analyses utilized three simultaneous evaluation metrics: the count of agents reaching the exit, exit times, and path costs.

Overall analysis

In the first study, the system was examined as a whole, accounting for the contributions made by both collaborators and defectors. The purpose was to figure out how group size affected the entire system. Fig 3.38 shows the number of exited agents for different group values as the collaborative factor f varies. Each colored tile in the legend indicates the number of groups Γ that the crowd has been split into. As the number of groups increases, the number of agents who reach the exit also increases, particularly when the crowd is separated into 50 or 100 groups, almost all of the agents in the crowd reach the exit when $0.6 \leq f \leq 0.9$. Crowd performance worsens as the Γ decreases, as well as when the crowd is entirely collaborative ($f = 1.0$), with the worst configuration being one in which all crowd agents belong to a single group. In conclusion, when the crowd is divided into several small groups that explore the environment at regular intervals and are separated from one another, many more agents reach the exit than when the same crowd is divided into a few very large groups or a single group that contains the entire crowd. The figures, Fig 3.39 and Fig 3.40 display the exit times and path costs, respectively. These metrics have been normalized based on the group success rate, which is the percentage of agents in a group that reaches the exit point. Interestingly, the crowd appears to exit faster when divided into a few groups with a large number of agents, but at the same time appears to find cheaper paths

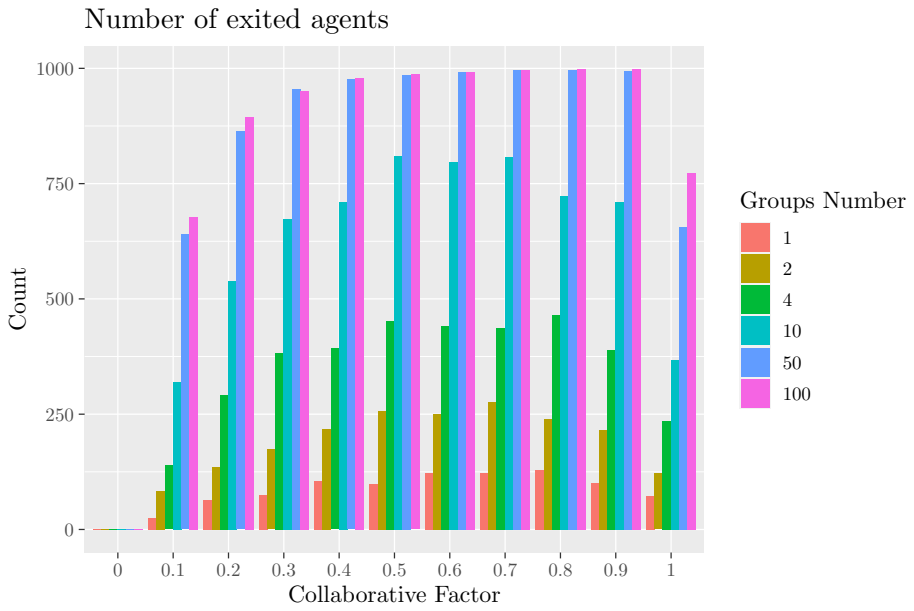


Figure 3.38: Overall number of exited agents

(as well as have many more agents that reached the exit) when divided into a large number of groups with few agents. This indicates that the exit times are obtained by a small number of agents and, therefore, evaluating all metrics simultaneously shows that the crowd's performance cannot be positively assessed when it is divided into a few groups with a lot of agents. Indeed, in this case, the path costs are worse and this suggests that the optimization cost process is driven primarily by the groups rather than the number of agents itself. Generally, except for the exit times, the more the number of groups into which the crowd is divided, the better its performance is. Furthermore, when the crowd is solely composed of collaborative agents ($f = 1.0$), its exit times and path costs are worse for all values of the groups' number.

Type analysis

The second study involved examining the system's performance by separately analyzing the performance of collaborators and defectors. The objective was to identify any disparities between the two types of perfor-

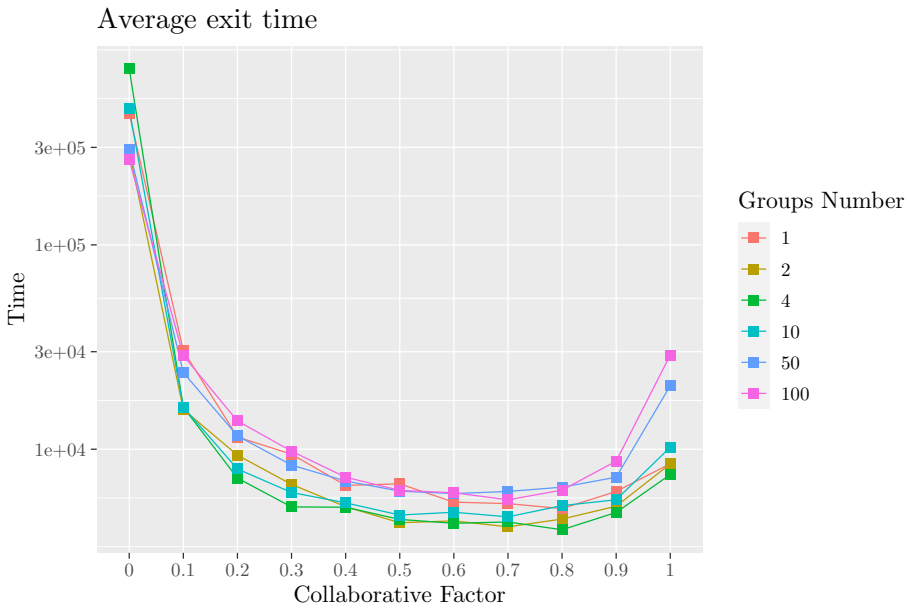


Figure 3.39: Overall exit time

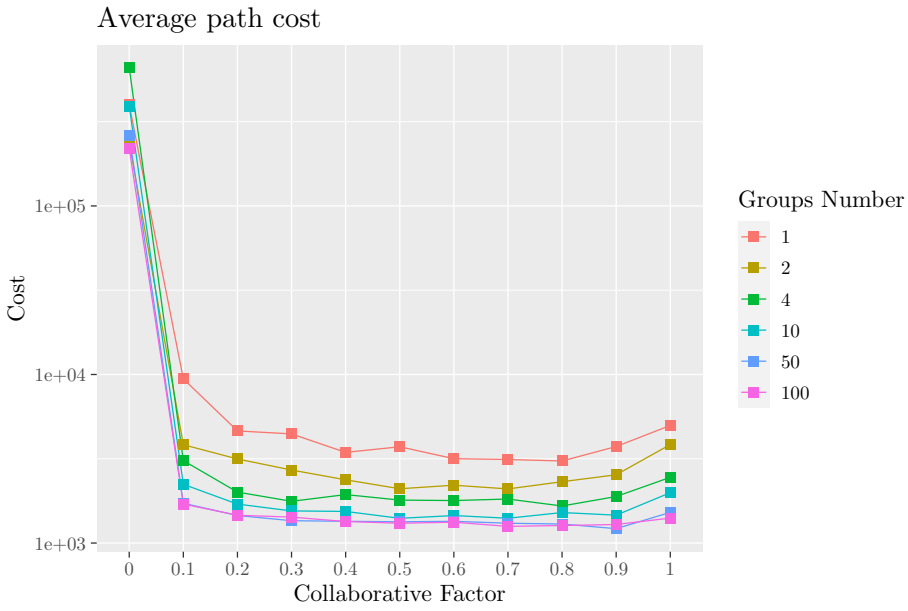


Figure 3.40: Overall path cost

mance and determine which type of agent would perform better. Fig. 3.41 displays the number of agents that exited, categorized by type, for various group values as the collaborative factor, f, changes. The number of exited collaborators and defectors both increase as the collaborative factor increases and reaches their maximum at different f values based on Γ. The most notable difference between the two is that collaborators' maximum is achieved at higher f values, and the trend is nearly linear for $\Gamma > 10$ with a slight drop at $f = 1.0$. In contrast, the defectors' maximum is attained at lower f values, and the trend is also mostly linear for $\Gamma > 10$, but with a decreasing trend. The two figures, Fig 3.42 and Fig 3.43, dis-

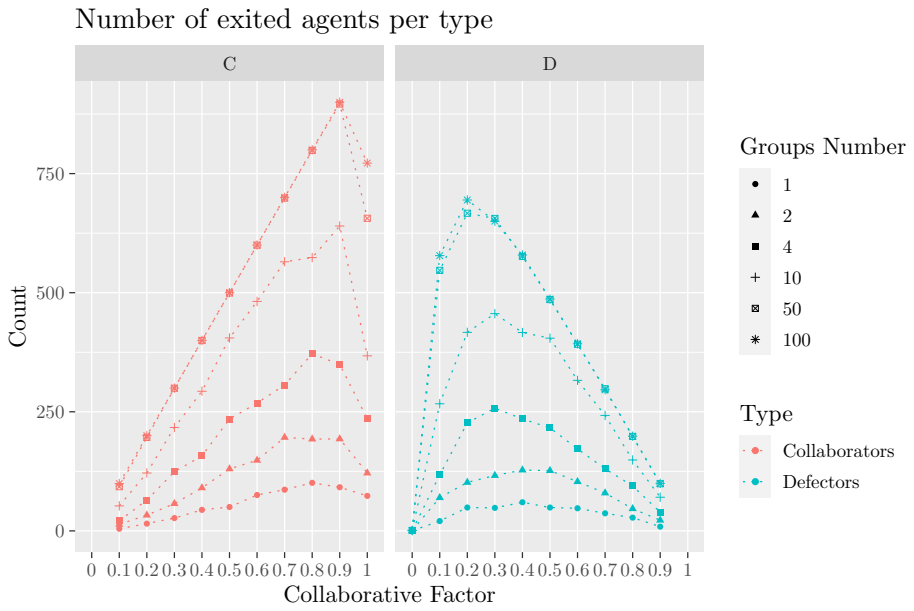


Figure 3.41: Number of exited agents per type C and D

play the exit times and path costs for both agent types at different values of f and Γ. The results confirm the overall analysis, which suggests that collaborators and defectors exit faster when there are fewer groups with many agents, while they find cheaper paths with more groups containing fewer agents. Collaborators exhibit different best exit times for different values of Γ, with $f = 0.9$ for $\Gamma \geq 50$ and $f = 0.7$ for $\Gamma \leq 50$. When $f = 1.0$, the performance worsens, as seen in the previous metric. Defectors show

smoother behavior, with exit times improving slowly as f increases. It is worth noting that there is a performance jump between $\Gamma = 50$ and $\Gamma = 10$, which may be due to a significant difference in values. These observations apply to the path costs as well, although the performance differences are less pronounced. Collaborators and defectors find better paths as Γ and f increase, and their behavior is more stable as Γ increases. However, in this case, collaborators do not perform better when $f = 1.0$

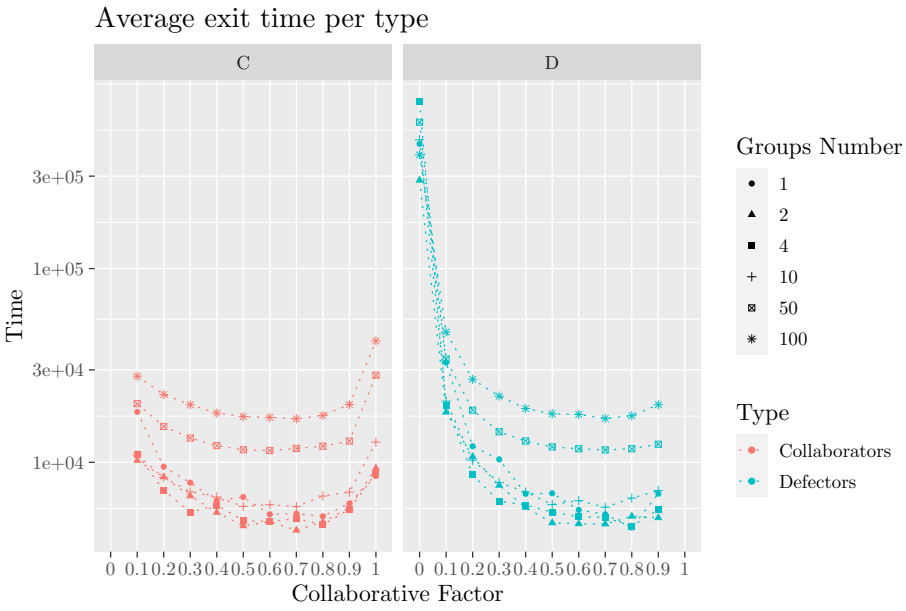


Figure 3.42: Exit time per type *C* and *D*

3.4.2 Conclusions

In conclusion, the study conducted with the third version of the model contributed to addressing the research question: *What impact does group size have on the behavior of collaborative and defector agents within the overall crowd?* Two kinds of analyses were conducted, each focusing on distinct aspects of the problem.

The overall analysis revealed that larger groups ($\Gamma = 2, 4$) generally lead to faster exit times. Thus, in situations prioritizing swift evacuation,

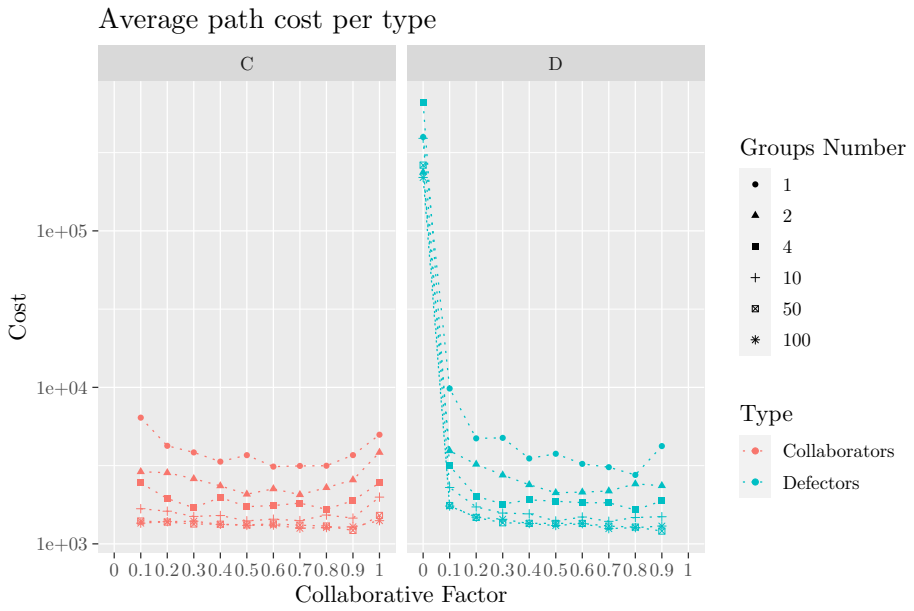


Figure 3.43: Path cost per type C and D

forming larger groups is advisable. Conversely, scenarios with numerous smaller groups ($\Gamma = 50, 100$) tend to facilitate a greater number of successful exits. Additionally, smaller groups excel in exploration, often discovering more cost-effective paths. Hence, when maximizing the number of evacuees is paramount, dividing the crowd into smaller groups can prove advantageous. Overall, the study corroborates that a combination of collaborative and non-collaborative elements in a crowd ($f \simeq 0.9$) yields superior performance compared to entirely collaborative crowds ($f = 1.0$).

The type analysis unveiled consistent trends for both collaborators and defectors individually. Small groups of both collaborators and defectors enhance exit success and path cost efficiency ($\Gamma = 50, 100$), while larger groups of both types facilitate faster exits ($\Gamma = 2, 4$). Collaborative agents consistently outperformed defectors in terms of successful exits, with collaborators demonstrating optimal performance in predominantly collaborative crowds ($f \simeq 0.9$). Defectors, on the other hand, excel in finding cheaper paths and achieving faster exits in predominantly collaborative

crowds ($f \simeq 0.9$), but exhibit enhanced exit success in defector-oriented crowds ($f \simeq 0.2$).

In conclusion, the findings from the third model underscore the interplay between group size and behavioral strategies. Larger groups prove advantageous for expediting evacuations, ensuring swift departures in scenarios prioritizing speed. Conversely, smaller groups emerge as better for achieving successful exits and uncovering cost-effective paths, particularly valuable when maximizing evacuee numbers is important. Moreover, the results confirm the interplay between collaboration and defection strategies, which significantly influences overall outcomes, with a balanced combination of both yielding superior results. Also, in this case, the presence of defector agents bolsters the group's resilience to uncertainty, facilitating more adaptive navigation in complex or unforeseen situations. Indeed, while collaborative agents prioritize information sharing to explore potential routes, defectors contribute alternative perspectives, thereby mitigating the risks of groupthink and ensuring diverse decision-making within the crowd.

3.5 Conclusions

In summary, the main objective of the second research line was to explore the interplay between cooperation and competition in social complex systems by comprehensively investigating this interplay through a coherent mapping of the ACO algorithm's rules into conceptual frameworks rooted in social dynamics. Subsequently, three distinct crowd models were developed, addressing the research questions:

1. *What is the interplay between cooperation and competition in social complex systems?*

   Cooperation and competition interact dynamically in social complex systems.

   In the initial study, an increase in the proportion of cooperative agents was found to correlate with higher average profit functions, even in groups primarily but not totally composed of cooperatives.

However, entirely cooperative or defector-dominated groups exhibited suboptimal performance, while a balanced mix of collaborators and defectors yielded superior outcomes. Collaborators tended to outperform defectors when in the minority, while defectors strategically leveraged collaboration when they formed the majority. Moreover, the presence of defectors paradoxically enhanced crowd efficiency, acting as a pruning mechanism and steering the crowd towards more efficient routes.

The second study transitioned into a comprehensive crowd simulation framework. Collaborators shared path cost information and could repair damage caused by defectors, while defectors did not share information and may cause harm by destroying edges or damaging vertices. The analysis confirmed that fully collaborative or fully defector crowds exhibited suboptimal performance, while a mix of collaborators and defectors yielded superior outcomes. Also here, collaborators tended to outperform defectors when in the minority, while defectors strategically leveraged collaboration when they formed the majority. They also demonstrated superior adaptability and effectiveness in scenarios where trace values were low or information was prioritized. Moreover, later-evacuating groups benefited from the information left by earlier groups, emphasizing the importance of collective knowledge sharing.

In the third study, the model was refined to enhance agents' ability to estimate edge weights, mirroring human-like reasoning more closely. Despite these adjustments, the model retained continuity with the previous version. Here, the research scope expanded to investigate how group size impacts the behavior of collaborative and defector agents within the crowd. Larger groups were found advantageous for expediting evacuations, while smaller groups excelled in achieving successful exits and identifying cost-effective paths. Additionally, it was confirmed also in this case that a balanced combination of collaboration and defection strategies led to superior outcomes, with defector agents enhancing the group's resilience to uncertainty and promoting adaptive navigation in complex situations. Collaborative agents prioritized information sharing for route ex-

ploration, while defectors offered alternative perspectives, mitigating groupthink risks and ensuring diverse decision-making within the crowd.

2. *Can the principles of the ACO algorithm be applied to simulate social dynamics?*

   Yes, the principles of the ACO algorithm can be applied to simulate social dynamics.

   By examining the core principles of the ACO algorithm, such as the proportional transition rule, reinforcement rule, and global updating rule, a parallel framework for modeling decision-making and environmental interaction within crowd simulations was established. Specifically, the proportional transition rule, which governs how ants make path choices based on factors like pheromone concentration and node visibility, was adapted to model how agents or evacuees make decisions regarding their routes in unfamiliar environments. This adaptation involved interpreting pheromones as traces left by agents and visibility as a measure of path desirability, which is communicated to neighboring paths to aid in identifying optimal routes. Similarly, the reinforcement rule, which governs the quantity of pheromone deposited by an ant upon traversing an edge, was employed to depict the interaction between agents and their environment. Agents leave traces as they navigate paths, reflecting the frequency of edge traversal and providing insights into their actions and the environmental state. Finally, the global updating rule, which dictates the gradual evaporation of pheromones over time, was utilized to simulate the impact of time on the environment. This rule controls the gradual reduction of agent traces over time, influencing how environmental information evolves with time. Therefore, by leveraging these principles, the ACO algorithm can effectively model and simulate various aspects of social dynamics within crowd simulations.

Chapter 4

Trust dynamics in multi-agent systems

4.1 Introduction

In the third research line, the focus shifted to a new question: What are the effects of trust dynamics in multi-agent systems? Addressing this, the agent-based model from the second research line, originally developed for collective behaviors, was modified to simulate social behaviors linked to trust. This model was then employed to understand the impacts of different behavioral strategies, particularly emphasizing trust and skepticism, especially in scenarios involving a robot assisting in navigation through unfamiliar environments. Before delving into detailed investigations and contributions, it's essential to establish the foundational context for a deeper exploration of this third research line.

Trust and the management of trust are pivotal concepts in establishing and maintaining satisfying relationships, extending beyond humans to various entities. These encompass relationships between humans and organizations, organizations and organizations, and even between humans and robots [45]. In general, entities establish a certain degree of trust to collectively achieve better results than what might be attainable individually.

In recent years, the use of robots and automated systems has surged,

not only in domestic settings but also in diverse sectors, including the medical field [77], mechanical engineering [31, 55] and in the study of human-robot interactions (HRI [9]). This is because robots have been designed not only to replace humans in dangerous tasks but also to assist them, make decisions in uncertain scenarios, or work as genuine teammates [45]. Defining trust unequivocally, however, is a complex task due to its inherently multidisciplinary nature. Psychologically, it is described as something an individual learns positively or negatively from experience, while sociologically, it is defined as the probability that a certain action corresponds to a benefit. In the economic realm, trust is studied from the perspective of investors cooperating in risky situations. Moreover, trust is dynamic, and identifying the factors influencing it depends heavily on the context considered [15].

Despite the difficulty of defining trust in human-robot interactions, the most accepted definitions in the field include *"the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability"* [47] and *"a belief, held by the trustor, that the trustee will act in a manner that mitigates the trustor's risk in a situation in which the trustor has put its outcomes at risk"* [84]. The latter definition recognizes two subjects: trustor i, who trusts to maximize their interest, and trustee j, who can impact the truster's interest [15].

Numerous studies have investigated the role of trust in human-robot interactions, leading to the development of diverse trust models. In works such as [68], researchers have aimed to identify the factors influencing trust in robots, while others, like [66, 65], delve into how trust fluctuates in response to errors committed by robots. The correlation between trust and performance in human-robot collaboration has been a focal point, giving rise to performance-based trust models.

An example of this is found in [96], where the authors reveal that team trust in robots positively impacts performance, while trust in humans is more closely tied to personal satisfaction. Additionally, [97] explores the impact of group identity and social attraction toward robots on team performance and viability, establishing social attraction as a mediator for the influence of group identity on these outcomes.

Expanding the perspective, [13] establishes a profound connection between the concept of trust and multi-agent systems (MAS). MAS, com-

putational systems with multiple decision-making agents, involves delegation, tasks, cooperation, exploitation, exchange, and teamwork, all intricately tied to trust. Trust emerges as a fundamental concept in MAS, shaping the dynamics of social interactions in both human and artificial intelligence realms. This is because, in both cases, the dynamics involve autonomous components acting and interacting to achieve goals in uncertain and dynamic environments. Consequently, trust and its ensuing systems are increasingly conceptualized, designed, and built using agent-based techniques, advocated as the natural computation model for such systems [64].

Building upon these foundational definitions and key concepts, this study draws inspiration from these related works. The proposed agent-based model aims to study the effects of trust toward robots considering the properties and dynamics of a multi-agent scenario where agents have to trust other virtual agents/robots in unknown environments. The model was built using the same framework as the crowd model defined in Chapter 3, consisting of a group of agents exploring a virtual environment to find the exit from a given entrance. In this environment, there is also an agent defined as a robot that explores the environment and provides information on the costs of the paths it has traversed. The goal of the agents is to find the exit in the shortest time possible, through paths that are as cost-effective as possible, and to exit as much as possible. To achieve this, they can adopt two strategies: either trust more in the information released by the robot than in what other agents do or trust more in what other agents do than in the information released by other agents. In other words, they can be either trusting or skeptical towards the robot. In this context, the trustor i is represented by the agents, while the trustee j is represented by the robot, and the situation of uncertainty is the exploration of the unknown environment.

The purpose of the model is to analyze whether, in situations characterized by uncertainty, such as the exploration of an unknown environment to find the exit, the assistance of a robot can improve the performance of the agents, helping them to exit more quickly, in greater numbers, and through less costly paths.

4.2 The trust model

In the model, a set of N agents navigate an environment, aiming to reach the exit by selecting the most cost-effective path. The environment is conceptualized as a weighted undirected graph $G = (V, E, w)$, wherein V represents the set of vertices, $E \subseteq V \times V$ signifies the set of edges, and $w\ colon V \times V \rightarrow \mathbb{R}^+$ functions to assign positive costs to each edge reflecting the challenge for agents in traversing these specific edges. In each scenario, agents start at the top-left node, aiming to reach the exit at the bottom-right. Let $A_i = j \in V : (i, j) \in E$ denote adjacent vertices to i, and $\pi^k(t) = (\pi_1, \pi_2, \ldots, \pi_t)$ represent the sequence of vertices visited by agent k at time-step t.

Within this environment, a robot actively assists agents in identifying optimal paths to the exit that are both economically viable and time-efficient. The robot accomplishes this by exploring the environment alongside the agents, strategically choosing paths of varying optimality based on its efficiency, and subsequently sharing information about the cost of a path after its traversal. The robot's efficiency r_e significantly impacts its path selections, with heightened efficiency directing the preference towards less costly paths. Agents and the robot spend time proportionate to the cost of traversing an edge. Consequently, an optimal path not only minimizes expenses but also requires less time for traversal.

Transitioning to the agents' behaviors, they may adopt two distinct strategies to find the exit:

- Trustful T: Agents trust and follow the robot's path cost information.
- Skeptical S: Agents neither trust nor follow the robot's information.

These strategies are derived from the proportional transition rule 1.1 of the crowd model. This rule shapes decision-making by weighing two types of information: the trace τ_{ij} unintentionally left by agents and the desirability η_{ij} of a path intentionally communicated by the robot. The trace is weighted by the parameter α, and the desirability is weighted by the parameter β.

In other words, under the Trustful Strategy (T), agents exhibit a predisposition to trust and align with the path cost information supplied by the robot. Characterized by $\alpha = 0.5$ and $\beta = 1.0$, trustful agents prioritize and heavily rely on the guidance offered by the robot.

On the contrary, the Skeptical Strategy (S) reflects a different scenario. Agents adopting this strategy neither trust nor follow the information provided by the robot. With parameters $\alpha = 0.5$ and $\beta = 0.0$, skeptical agents assign more weight to information acquired from other agents in the environment. In essence, these agents lean towards a more independent decision-making process, relying on the collective wisdom and experiences of their peer agents rather than solely relying on the robot's directives.

Essentially, the study aims to explore the influence of a robot's presence on the overall performance of a group of agents employing distinct strategies. Three key evaluation metrics are used to evaluate agent performance in this context: the number of exited agents, exit times, and path costs. Through the application of these evaluation metrics, the study aims to elucidate how interactions between agents and the robot contribute to their collective performance. This exploration of trust, skepticism, and decision-making dynamics forms a comprehensive framework for understanding the impact of robotics in the context of navigation within unfamiliar environments.

4.3 Experiments and results

For the experiments, three graphs C1, C2, and C3 were utilized, each comprising $|V| = 400$ nodes. The respective edge counts for these graphs are $|E| = 511$, $|E| = 561$, and $|E| = 599$. Within each graph, nodes connect to up to four neighbors, and edge weights are real numbers selected uniformly from the range $]0, 10]$. C1, C2, and C3 environments were purposefully constructed to provide the agents and the robot with various challenges, progressing from a simpler scenario in C1 to a more intricate situation in C3. This intentional variation enables a thorough assessment of the agent's adaptability and performance across different levels of environmental complexity.

The simulations involved a single group, $\Gamma = 1$, consisting of $N = 100$ agents. The trust factor denoted as $f \in [0, 1]$, governed the ratio of trustful (T) to skeptical (S) agents. For these experiments, $f = 0.5$ was set, resulting in an equal distribution of $N_T = 50$ trustful and $N_S = 50$ skeptical agents. A trace amount specific to their kind denoted as $K_{T,S} = 0.1$, was released by each agent, with an initial value set at $\tau_{T,S}(t = 0) = 0.1$. It is noteworthy that time impact was not incorporated into this model.

The robot, on the other hand, releases information $\eta_{ij} = \frac{1}{w_{ij}}$ after crossing a path, with w_{ij} cost of the crossed path.

For the experiments, different kinds of situations were considered:

- **No Robot**: Agents independently explore the environment, serving as a baseline for comparison.
- **Immediate Start**: The robot starts exploration simultaneously with agents.
- **Early Start**: The robot begins exploration before agents.

Due to computational constraints, all agents had limited time to find the exit, calculated differently depending on the situation under investigation. In the case of No Robot and Immediate Start, it is $T_{\max} = |V| \times \frac{N}{2}$, while in the case of Early Start, it is $T_{\max} = |V| \times \frac{N}{2} + R_t$, where R_t is defined as $|V| \times \frac{N}{2}$ and represents the time the robot has to explore the environment before agents begin their exploration. This was done to ensure that agents had an equal amount of time to navigate the environment in all three configurations.

To assess the influence of the robot's presence, experiments were conducted to explore the relationship between agent performance and both the type of analysis and the efficiency of the robot. Two specific values of robot efficiency, namely $r_e = 0.2$ and $r_e = 0.8$, were the focus of this study. It should be emphasized that the robot's path selection follows a probabilistic approach, wherein the likelihood of opting for less costly paths increases with higher efficiency and decreases with lower efficiency. The selection of these efficiency values—0.2 denoting low efficiency and 0.8

denoting high efficiency—was intentional to avoid extreme values, preventing the portrayal of overly specific or unrealistic scenarios. Instead, a deliberate choice was made to utilize values at the limits, maintaining a balance for a thorough evaluation without overly complicating the experimental design. This approach aimed to represent plausible real-world situations. Each efficiency level underwent 100 independent simulations, contributing to a comprehensive assessment of their impact on the experimental outcomes.

The upcoming visualizations showcase graphs for various evaluation metrics. Specifically, Fig. 4.1 illustrates the number of exited agents, Fig. 4.2 displays exit times, and Fig. 3.40 represents path costs. These graphs are thoughtfully grouped to improve clarity and facilitate comparisons across different scenarios. Each plot consists of five columns: No robot, Immediate start with robot efficiency $r_e = 0.2$, Immediate start with robot efficiency $r_e = 0.8$, Early start with robot efficiency $r_e = 0.2$, and Early start with robot efficiency $r_e = 0.8$.

At first glance, it appears that in all three scenarios, both trustful and skeptical agents exit in nearly equal numbers, exhibiting comparable exit times and path costs in the absence of a robot. However, a notable shift occurs when introducing a robot, particularly with an early start and a robot efficiency value of 0.8, especially evident in scenario C3. To validate the statistical significance of these observations and affirm the impact of a robot's presence, t-tests were conducted on exit times and path costs. The t-test is a statistical tool that evaluates observed differences between group means. It encompasses formulating null and alternative hypotheses, calculating a t-statistic based on group mean differences, and determining degrees of freedom. In this context, the means represent trustful and skeptical agents for each evaluation metric. The null hypothesis posits no differences between the groups, corresponding to the baseline without a robot. The alternative hypothesis suggests statistically significant differences between the two group means. Charts displaying statistically significant results according to the t-test are marked with (*).

Tables 4.1 and 4.2 represent the statistical analysis respectively for the path costs and the exit times in scenarios C1, C2, and C3. The statistical measures include t-values (t), degrees of freedom (df), *p-values*, and 95% confidence intervals (CI).

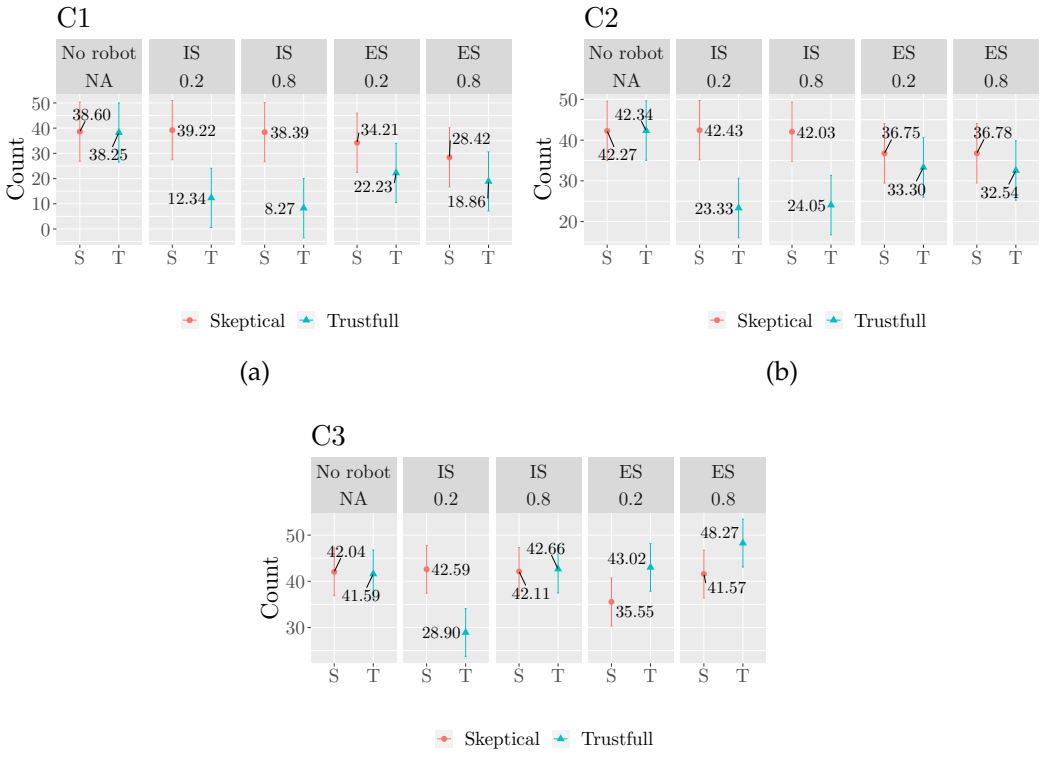


Figure 4.1: Number of exited agents for scenarios C1 (a), C2 (b), and C3 (c).

4.3.1 Results analysis

Considering the path costs, the results reveal consistent patterns across scenarios, indicating that the presence of a robot, particularly with immediate starts, significantly influences path costs. The extremely low *p-values* in scenarios with the robot suggest a highly significant impact on the measured output, further supported by wide confidence intervals that do not include zero. In contrast, scenarios without the robot generally show minimal impact on path costs, as evidenced by higher *p-values* and narrower confidence intervals that include zero. This underscores that in the absence of the robot, the trustful and skeptical agents behave similarly, confirming the validity of the null hypothesis.

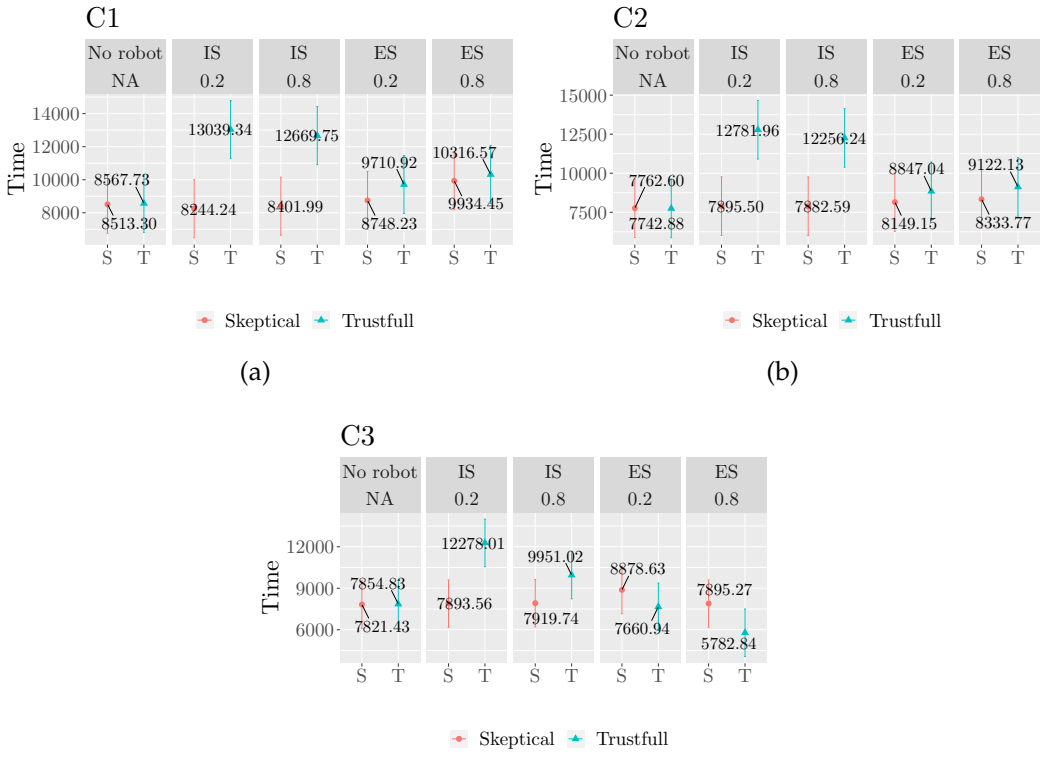


Figure 4.2: Exit times for scenarios C1 (a), C2 (b), and C3 (c).

The examination of exit times demonstrates distinct trends in response to various conditions. In C1, the introduction of the robot with different start modes, especially Immediate start 0.2 and Immediate start 0.8, significantly and adversely affects the measured output. Notably, an Early start of 0.8 has a less pronounced impact. Similarly, for C2, negative effects are observed with Immediate start conditions, while the influence of Early start conditions is less conspicuous. Finally, in C3, the inclusion of robots with Immediate start modes significantly negatively influences the output, whereas Early start 0.2 and Early start 0.8 appear to have a positive impact on the measured output.

In summary, the addition of a robot with various start modes has a significantly positive impact on the measured output especially in sce-

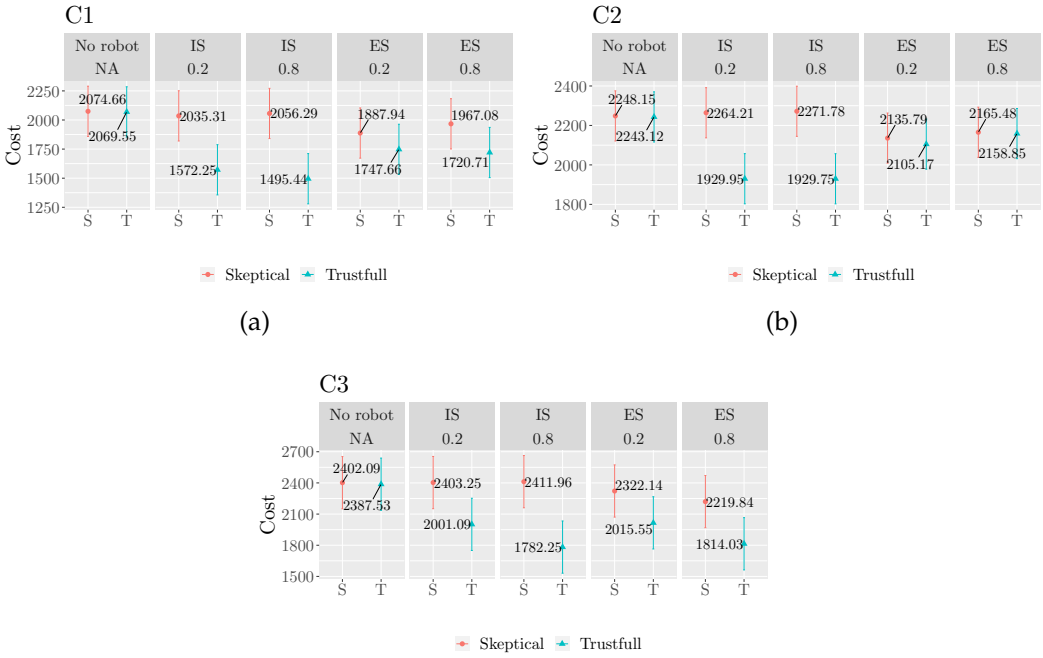


Figure 4.3: Path cost for scenarios C1 (a), C2 (b), and C3 (c).

nario C3. The differences between the Skepticals and Trustfuls agents are particularly pronounced when the robot is early initiated (both with 0.2 and 0.8 probabilities). Taking into account even the number of agents reaching the exit, Scenario C3 appears to exhibit the most favorable performance.

Fig 4.1 represents the number of exited agents across all scenarios. It can be observed that the presence of a robot with an immediate start does not seem to significantly improve agent exit rates, especially in scenarios C1 and C2. This suggests that in such scenarios, trusting the robot may not be advantageous. In C3, this holds true only when the robot efficiency is 0.2. When the efficiency rises to 0.8, there is a noticeable improvement in the performance of trustful agents. This trend becomes more pronounced in the case of an early start with both robot efficiency values, 0.2 and 0.8. Trustful agents significantly outperform skeptical ones when the robot efficiency is 0.8, reaching the maximum number of exited agents.

PATH COST						
Scenario	**Metric**	**NR**	**IS 0.2**	**IS 0.8**	**ES 0.2**	**ES 0.8**
C1	t	0.36236	26.676	28.797	8.984	15.323
	df	7680.8	2587.1	1477	4951.5	4345.4
	p-value	0.7171	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$
	95% CI	-22.53 32.74	429.02 497.10	522.65 599.05	109.67 170.89	214.85 277.89
C2	t	0.33418	21.163	21.622	1.9036	0.41572
	df	8458.9	5620.7	5677.4	6955.9	6894.7
	p-value	0.7383	$< 2.2e^{-16}$	$< 2.2e^{-16}$	0.057	0.6776
	95% CI	-24.50 34.57	303.30 365.23	311.02 373.04	-0.91 62.15	-24.63 37.88
C3	t	0.85137	23.943	40.731	17.521	25.521
	df	8352.9	6907.9	8147.1	7715	8739.2
	p-value	0.3946	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$
	95% CI	-18.96 48.08	369.23 435.08	599.40 660.02	272.28 340.88	374.64 436.98

Table 4.1: Path costs for scenarios C1, C2, and C3

EXIT TIME						
Scenario	**Metric**	**NR**	**IS 0.2**	**IS 0.8**	**ES 0.2**	**ES 0.8**
C1	t	-0.45564	-30.705	-21.297	-6.5318	-2.3728
	df	7675.6	2332.7	1200.3	4703.4	4031.3
	p-value	0.6487	$\ll$ 2.2e-16	$\ll$ 2.2e-16	7.19E-11	0.0177
	95% CI	-288.5784 179.7279	-5101.341 -4660.926	-1925.3754 -764.22262	-4488.853 -3874.592	9934.455 10316.566
C2	t	0.17784	-40.482	-35.622	-5.4684	-6.1275
	df	8458.1	5358.1	5329.8	6888.4	6771.8
	p-value	0.8589	$\ll$ 2.2e-16	$\ll$ 2.2e-16	4.70E-08	9.43E-10
	95% CI	-197.6513 237.0926	-5123.094 -4614.349	-948.0766 -1040.578	-4649.819 -4132.95	-447.7142 -536.153
C3	t	-0.29843	-37.308	-19.432	9.9146	20.944
	df	8343.8	6523.2	8297.3	7502	8170.8
	p-value	0.7654	$\ll$2.2e-16	$\ll$2.2e-16	$\ll$2.2e-16	$\ll$2.2e-16
	95% CI	-252.8402 186.0260	-4614.826 -2236.198	976.9339 1458.4489	1914.71 2310.14	-4154.074 -1826.373

Table 4.2: Exit times for scenarios C1, C2, and C3

About the exit times, depicted in Fig 4.2, in the case of an immediate start with robot efficiency values of 0.2 and 0.8, the presence of a robot does not yield significant improvements in agent exit times across all three scenarios. This implies that, under this specific immediate start configuration, even with an increase in robot efficiency, the performance of agents, especially trustful ones, remains largely unaffected. In the case of an early start with both robot efficiency values, while the performance of skeptical agents worsens, trustful agents improve their exit times. In particular, in scenario C3, trustful agents outperform skeptical ones, exiting faster for both values of robot efficiency.

For the path costs, in Fig 4.3, the presence of a robot significantly improves trustful agents' path costs in all three scenarios and configurations. Skeptical performance remains the same or even worsens (compared to

the no-robot configuration) in the case of an immediate start with robot efficiency values, while they improve their performance in the case of an early start with both robot efficiency values. However, their performance remains worse than the trustful ones.

Examining all three evaluation metrics concurrently, it's evident that, in the absence of a robot, trustful and skeptical agents demonstrate similar performance, exiting with comparable exit times and path costs. Notably, the model excels in scenario C3 and the early start 0.8 configuration, where trustful agents exhibit optimal performance.

The immediate start configuration revealed disparities between exit times and path costs, which can be attributed to the inherent efficiency-driven decision-making process of the robot and its initiation timing. The robot prioritizes path cost optimization and navigation efficiency, indirectly influencing exit times. The efficiency parameter, denoted by values of 0.2 and 0.8, guides the robot's path selection strategy. A higher efficiency setting consistently results in significantly lower path costs, especially for trustful agents, aligning with the expectation of cost-effective choices. However, the impact on exit times unfolds differently. Immediate starts, despite enhancing path costs, adversely affect exit times, particularly for trustful agents. This suggests that the immediate initiation of a more efficient robot may not necessarily expedite the exit process. On the contrary, early starts positively influence both path costs and exit times.

In essence, while the robot consistently contributes to reporting economical paths through efficient selection, its influence on exit times hinges on finding the right equilibrium between efficiency and initiation timing.

4.4 Conclusions

In summary, the third research line aimed to investigate the impacts of trust and skepticism in scenarios where a robot assists in navigating unfamiliar environments. To address this, modifications were made to the agent-based model from the second research line to simulate social behaviors related to trust, tackling the research question:

1. *What are the effects of trust dynamics in multi-agent systems?*

Trusting a robot, especially with higher efficiency, enhanced agent performance, leading to expedited exits through more efficient paths.

The effects of trust dynamics in multi-agent systems were investigated using an agent-based model, focusing on how trust and skepticism impacted the performance of virtual agents navigating unfamiliar environments with robot assistance. The study aimed to assess whether trusting a robot, with varying levels of efficiency, could enhance overall agent performance in uncertain situations. Three scenarios were analyzed: No Robot, where agents explored independently; Immediate Start, where the robot began exploration simultaneously with agents; and Early Start, where the robot initiated exploration before agents. Results showed that trustful agents outperformed skeptical ones, particularly with a high-efficiency robot and an early start. The presence of a robot significantly improved agent performance, especially in complex scenarios, with higher robot efficiency leading to quicker exits and improved path costs.

Conclusions

The central inquiry driving this doctoral project has been the role of competition within complex systems that heavily rely on cooperation. This fundamental question stemmed from the observation that various seemingly disparate systems share fundamental similarities. These systems, ranging from societal structures to global climates and ecosystems, exhibit complex behaviors driven by interactions among their constituent elements and with the surrounding environment. Network science has emerged as a powerful tool for understanding and addressing the challenges posed by these complex systems, particularly through the framework of optimization problems. In this framework, systems are viewed as networks where nodes represent the system's elements and links are the relationships between them.

Traditionally, exact methods were employed to tackle optimization problems. However, the exponential growth of network sizes and the realization that real networks are dynamic and with non-trivial topological characteristics have spurred the adoption of approximate methods. In this context, Swarm Intelligence algorithms have emerged as promising solutions. Rooted in principles of self-organization and decentralized control, draw inspiration from various natural phenomena, such as the foraging behavior of ants, the flocking patterns of birds, and the social dynamics of bees. By mimicking these collective behaviors, Swarm Intelligence algorithms offer scalable and adaptive solutions to optimization problems within dynamic networks. Their decentralized nature allows for robustness against perturbations and dynamic changes, making them

particularly well-suited for addressing real-world challenges across diverse domains. Their efficacy lies in the emergence of collective behaviors, where macroscopic and global properties of the system cannot be deduced from individual elements' characteristics.

However, as the investigation progressed, it became evident that cooperative and competitive dynamics coexist within these systems. Insect societies, initially perceived as cohesive groups, were found to harbor conflicts of interest among members. Similarly, complex animal species and human contexts exhibit a delicate balance between cooperation and competition. This realization underscored the inseparable nature of cooperation and competition, culminating in the central research question: What role does competition play within complex systems that also rely on cooperation? This question has guided the development of the research project into three interconnected lines, each exploring distinct domains arising from this foundational inquiry.

The first research line addressed the potential contribution of competition to optimization in the context of optimization problems. This led to the development of a tailored Ant Colony Optimization algorithm (ACO) for dynamic networks. By implementing cooperative and competitive ant colonies in virtual environments, the aim was to assess whether competitive dynamics could enhance the ACO algorithm's efficiency.

Transitioning to the social aspects of collective behaviors, the second research line investigated the interplay between cooperation and competition in social complex systems. This prompted the creation of a crowd social model, inspired by ACO algorithm principles, to simulate social dynamics within crowds. Through various analyses, the goal was to understand the impacts of different behavioral strategies, particularly focusing on cooperation and defection, within crowd contexts.

The third research line introduced a different perspective by adapting the agent-based model developed in the second research line. This adjustment aimed to simulate social behaviors associated with trust. This led to an investigation into the effects of trust dynamics in multi-agent systems, emphasizing trust and skepticism, particularly in the context of a robot aiding navigation through unfamiliar environments.

In each research line, the proposed models were implemented using an agent-based approach, a computational framework designed to sim-

ulate the actions and interactions of autonomous agents. This approach facilitated the exploration of agent behaviors and their outcomes in complex systems. Specifically, NetLogo was chosen as the programming language and Integrated Development Environment (IDE) for its suitability in implementing agent-based models.

Optimization Problems Research Line

In summary, the first research line aimed to explore the impact of competitive dynamics on optimization algorithms, specifically focusing on the Ant Colony Optimization algorithm. Through the development of two distinct ant models using an agent-based approach, the overarching question guiding this investigation was: could competition contribute to the optimization of outcomes in the context of optimization problems? The initial ant model demonstrated that competition, when strategically integrated, can enhance algorithmic efficiency. The introduction of competitive ants led to positive effects, notably improving the algorithm's success rate. Their strategic actions prompted the colony to adapt and explore alternative paths, resulting in overall performance enhancement. Similarly, in the second ant model, the presence of competitive ants positively influenced efficiency, especially in environments rich with trace information. However, this also highlighted the intricate interplay between cooperative and competitive dynamics, particularly emphasized in scenarios with high trace information levels. This insight underscores the importance of competition, especially in environments where the algorithm can strategically adapt and optimize its performance based on environmental characteristics. In challenging the traditional approach, the proposed models revealed that competition, when strategically integrated, can enhance algorithmic efficiency. Additionally, the agent-based approach proved to be a powerful tool for exploring algorithm efficiency across various scenarios, highlighting the synergistic relationship between competition and the agent-based approach in optimizing outcomes.

Collective Behaviors Research Line

In summary, the second research line aimed to explore the interplay between cooperation and competition in social complex systems by mapping the rules of the ACO algorithm into conceptual frameworks rooted in social dynamics. Three distinct crowd models were developed to address the research questions regarding this interplay. The initial study revealed that an increase in the proportion of cooperative agents correlated with higher average profit functions, leading to superior outcomes in groups with a balanced mix of collaborators and defectors. Defectors enhanced crowd efficiency, acting as a pruning mechanism and steering the crowd towards more efficient routes. Subsequent analyses within a comprehensive crowd simulation framework reaffirmed the benefits of a balanced combination of collaboration and defection strategies, with defector agents enhancing group resilience to uncertainty and promoting adaptive navigation in complex situations. Collaborative agents prioritized information sharing for route exploration, while defectors offered alternative perspectives, mitigating groupthink risks and ensuring diverse decision-making within the crowd. Furthermore, the research demonstrated that the principles of the ACO algorithm can effectively simulate social dynamics by adapting core principles such as the proportional transition rule, reinforcement rule, and global updating rule. This adaptation provided a parallel framework for modeling decision-making and environmental interaction within crowd simulations, showcasing the versatility and applicability of the ACO algorithm in capturing cooperative and competitive behaviors in social complex systems.

Trust dynamics in multi-agent systems research line

In summary, the third research line aimed to investigate the impacts of trust and skepticism in scenarios where a robot assists in navigating unfamiliar environments. Modifications were made to the agent-based model from the second research line to simulate social behaviors related to trust, addressing the research question: What are the effects of trust dynamics in multi-agent systems? Trusting a robot, especially with higher efficiency, enhanced agent performance, leading to expedited exits through more efficient paths. The study employed an agent-based model to ex-

plore how trust and skepticism influenced the performance of virtual agents navigating unfamiliar environments with robot assistance. Three scenarios were analyzed: No Robot, Immediate Start, and Early Start. Results indicated that trustful agents outperformed skeptical ones, particularly with a high-efficiency robot and an early start. The presence of a robot significantly improved agent performance, especially in complex scenarios, with higher robot efficiency leading to quicker exits and improved path costs.

Limitations, Future Directions, and Practical Applications

While the exploration of optimization in dynamic networks, collective behaviors in crowds, and trust in multi-agent systems through agent-based models provides valuable insights, it's crucial to recognize its limitations, delineate future research directions, and contemplate practical applications.

In the **Optimization problems research line**, the ant models, while illustrative, have primarily been evaluated on networks of moderate size and complexity. Notably, these evaluations involve computer-generated networks rather than real-world ones. Real networks have large dimensions, follow power-law degree distributions, and possess non-trivial topological characteristics. The applicability of the findings to diverse real-world environments, beyond maze-like structures, may influence the generalizability of results across traditional optimization benchmarks. Additionally, while the agent-based approach can be a powerful tool that enables the easy exploration of algorithm efficiency across a variety of scenarios, it may not be the ideal tool for analyzing instances of large dimensions due to the high computational cost inherent in this approach.

In light of this, future directions in this research line include, firstly, translating the proposed framework into another programming language and testing it on benchmark instances. Moreover, concerning the dynamics of the model, future directions could explore dynamic adaptation mechanisms by introducing ants capable of adjusting their cooperative or competitive behaviors in real-time. This dynamic adaptation, inspired by machine learning concepts, would enable the colony to continuously optimize its strategies in response to evolving challenges. Additionally, prac-

tical applications could involve applying these models in fields such as supply chain management, where dynamic environmental changes and resource allocation are critical factors. Furthermore, exploring hybrid approaches with other optimization algorithms, such as machine learning or genetic algorithms, may unlock new avenues for increased efficiency and adaptability in diverse contexts. This could find applications in logistics, where optimizing routes and resource utilization are fundamental.

In the **Collective behaviors research line**, the developed crowd models, while striving to achieve a balance between realism and simplicity, may have oversimplified the complexity of real-world scenarios. On one hand, although the ACO algorithm can effectively model and simulate various aspects of social dynamics within crowd simulations, its current implementation considers agents making decisions based solely on rational factors, overlooking the emotional and irrational dimensions inherent in human decision-making. While translating pheromones into traces and visibility into path desirability captures certain aspects, it may not fully encompass the richness of human decision-making processes. Additionally, while the obtained results, although partially counterintuitive, align with expectations derived from the literature, they were obtained with limited parameter sets. Sensitivity analysis of parameters has revealed more complex scenarios regarding the effectiveness of competition in social systems. Furthermore, how collaborators and defectors are modeled may fall short of representing the full spectrum of crowd behavior variability in dynamic and unpredictable environments. Future directions in this research line could first explore other dimensions inherent in human decision-making, such as elements of democratic, emotional, and irrational decision-making. Additionally, future research might delve into dynamic adaptation mechanisms, where agents can dynamically adjust their behaviors in response to real-time environmental changes. Integrating environmental heterogeneity, such as obstacles or dynamic changes in the network structure, would further enrich the models, offering insights into how crowds adapt in diverse and challenging environments. Real-world validation remains a crucial aspect of future research endeavors. Implementing and validating the models in physical environments or with real crowds would bridge the gap between simulations and practical applications, ensuring their effectiveness and applicability in real-world

scenarios. This empirical validation process is essential for refining the models and gaining confidence in their predictive capabilities. Furthermore, practical applications of these models include, for instance, optimizing evacuation plans in emergency scenarios, urban planning, and crowd management. By leveraging these models, spaces can be designed to facilitate smoother crowd movement, contributing to improved city infrastructure and enhancing overall safety and efficiency.

In the **Trust dynamics in multi-agent systems research line**, although the trust model provides a structured approach and the obtained results are plausible and in line with common sense expectations, it may not adequately capture such intricate interactions. First of all, the deliberate selection of robot efficiency values—0.2 representing low efficiency and 0.8 representing high efficiency—was a strategic decision aimed at avoiding extreme values and ensuring a realistic portrayal of plausible scenarios. However, this intentional constraint may restrict the applicability of the model to scenarios with efficiency values beyond this range. Trust dynamics in real-world scenarios, moreover, can be influenced by a myriad of factors beyond the efficiency of the robot, such as communication methods, reliability, and adaptability, while in the model there is a binary distinction between trusting the robot or fellow agents, that oversimplifies these dynamics. Future directions in the research line of trust multi-agent systems could explore the refinement and application of trust models in increasingly dynamic and unpredictable environments. Integrating real-world factors, such as environmental changes, unexpected obstacles, varying levels of robots' reliability or agents' level of trust could enhance the realism of the model. The practical applications of this research extend beyond environmental exploration scenarios. Trust models developed in this context could be instrumental in shaping trust dynamics in robots in fields like collaborative manufacturing or healthcare settings, where robots assist medical professionals or interact with patients. Furthermore, these trust models might find relevance in autonomous vehicles, optimizing the collaboration between human drivers and automated systems for safer and more efficient transportation. In essence, future research could focus on adapting the model to diverse real-world scenarios, ensuring its applicability across various industries and domains.

Concluding Remarks

In conclusion, this doctoral project has delved into the intricate interplay between cooperation and competition within complex systems, spanning from optimization problems to collective behaviors. Through the lens of network science and the utilization of Swarm Intelligence algorithms, particularly Ant Colony Optimization, the research has elucidated how competition can enhance optimization outcomes in dynamic environments. Moreover, the exploration of collective behaviors in crowds and the investigation of trust dynamics in multi-agent systems have provided insights into the dynamics of collective behaviors. Despite inherent limitations and challenges, the developed models aim to offer new avenues for future research and practical applications, contributing to the advancement of understanding complex systems.

Personally, embracing this doctoral challenge in computer science has been one of the biggest challenges of my life so far, especially considering my background, which is quite different — I graduated in physics. Engaging with the topics described in this thesis has undoubtedly helped me grow both as a person and as a researcher. It has provided me with a deeper understanding of what it truly means to conduct research. Moving forward, I am grateful for the opportunity to have embarked on this journey, and I hope to continue exploring new frontiers of knowledge and contributing to the world of research.

Appendix A

Supporting Materials

In this appendix, supplementary materials are provided in the form of images depicting various versions of the models presented in each research line. These visual aids are intended to enhance the comprehension of the model's functionality and dynamics discussed throughout the thesis.

Fig. A.1 illustrates the graphical interface of the first Ant Model developed within the first research line. The model comprises three tabs: the interface, where the simulation happens, Info, where model-related information is entered, and Code, containing the model's code. Within the Interface tab, on the left side, various buttons and sliders facilitate the initialization of the simulation environment. Notably, the Setup button initializes all necessary simulation variables, while the Go button initiates the simulation. Below these buttons, a series of sliders allow for selecting the population size, the amount of pheromone deposited by the ants, the evaporation rate, the number of ant generations, the fraction of cooperative ants, and the number of simulations. In the central part, the network on which the ants move is visible. Two types of ants are identifiable: red ones, representing cooperative ants, and blue ones, representing competitive ants. The entrance to the environment is depicted as an orange circle on the left, while the exit is represented by a blue circle on the right. The numbers on the links represent the amount of pheromone on that link at a specific time, while the fire symbols represent nodes blocked by competitive ants. Finally, on the right side, there is an output window where the number of resources gained by the ants is cumulatively recorded and

counted only at the end of the simulation.

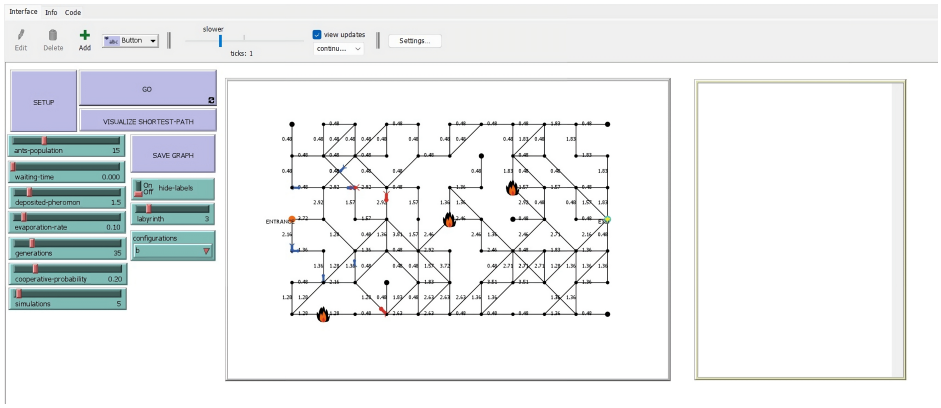


Figure A.1: Interface of the first ant model

Fig. A.2 represents the graphical interface of the third crowd model developed within the second research line. Also in this model are recognizable the setup button to initialize the simulation environment and the start button initiates the dynamics. Additionally, several elements are identifiable for inputting parameters defining the population size, the number of groups to be divided, the value of the trace increment for agents, the trace evaporation rate, as well as the values of the alpha and beta parameters. Sliders allow for defining not only the composition of the crowd, particularly regarding the cooperative factor—how many cooperative agents and defectors are present in the crowd—but also the probabilities of node damage and repair within the network. Furthermore, a series of monitors illustrate the scenario name, the current simulation progress out of the total simulations, as well as the number of nodes and links. In the central part, there is the environment representing the simulation's setting. Two types of agents are recognizable: the red ones representing defectors and the green ones representing collaborators. They move within the environment to reach the exit, positioned in the bottom right corner, while the entrance is on a node slightly obscured by a defector above it, located on the left side of the environment. At the bottom, there is an output window recording the characteristics of each agent that reaches the exit, along with the time taken to reach the exit and the path cost. On the far right, there is a panel allowing for

an investigation into the variables defined for each agent, including the lists of visited nodes and links, the *who* identifier, the shape defining the agent's form, and the strategy—here denoted as C, indicating in this case a collaborator.

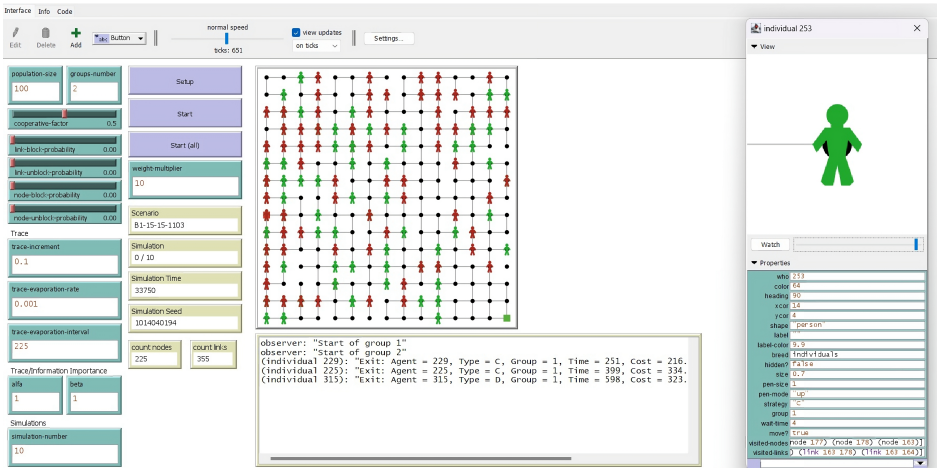


Figure A.2: Interface and features of the agents of the third crowd model

Finally, fig. A.3 illustrates the interface of the trust model developed within the third research line. Also here, on the left side, the graphical interface features several elements, including the setup button to initialize the simulation environment and the start button to begin the simulation. Sliders are available to adjust parameters such as the number of rows and columns defining the network. Additionally, sliders for alpha trustful, beta trustful, alpha skeptical, and beta skeptical enable the modeling of trustful and skeptical agents, along with another slider defining robot efficiency. Panels allow for the selection of the total population, the number of simulations, and traces left by both skeptical and trustful agents. Monitors display the maximum simulation time and the current simulation progress. In the center, the environment where agents move is presented. Two types of agents are identifiable: red-colored agents representing the Skepticals and green-colored agents representing the Trustfuls. The robot, depicted in blue and labeled with its name, is also visible. The entrance to the environment is situated at the top left, while the exit is positioned at the bottom right. Additionally, on the right side, an output window

records information for each agent reaching the exit, including the identifier, agent type, and other details such as the arrival time and path cost. These metrics are the ones used for the proposed analyses.

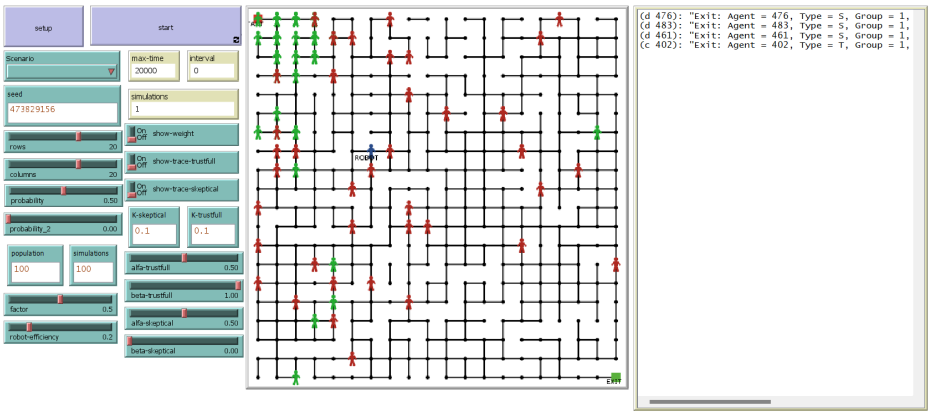


Figure A.3: Interface of the trust model

In all developed models and for all considered metrics, the analyses proposed throughout the thesis were conducted following a standardized procedure. The results obtained from the output windows were saved in .txt files. Subsequently, all .txt files corresponding to simulations with a specific configuration underwent preprocessing using a Python script. This preprocessing generated a single CSV file containing the results of all simulations conducted with that particular configuration. Finally, this CSV file was loaded into R software enabling the execution of all proposed analyses.

Bibliography

[1] J. Smith A. E. Eiben. From evolutionary computation to the evolution of things, 2015.

[2] John Acorn. Complex Adaptive Entomology. *American Entomologist*, 64(1):63–64, 03 2018.

[3] Kayvan Aghabayk, Nasser Parishad, and Nirajan Shiwakoti. Investigation on the impact of walkways slope and pedestrians physical characteristics on pedestrians normal walking and jogging speeds. *Safety Science*, 133:105012, 2021.

[4] Khaled Akka and Farid Khaber. Mobile robot path planning using an improved ant colony optimization. *International Journal of Advanced Robotic Systems*, 15(3):1729881418774673, 2018.

[5] Shams Al-Amin, Emily Z. Berglund, and Kelli L. Larson. *Complex Adaptive System Framework to Simulate Adaptations of Human-Environmental Systems to Climate Change and Urbanization: The Verde River Basin*, pages 1819–1825. Wayne C. Huber, Ph.D., P.E., D.WRE, Portland, Oregon, 2014.

[6] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, Jan 2002.

[7] Víctor M Albornoz and Cristian M Canales. A non-linear optimization model for obtaining a total allowable catch quota of the

chilean jack mackerel fishery. *Journal of Applied Operational Research*, 5(4):153–163, 2013.

[8] Aliyu Aliyu and Neyre Tekbiyik-Ersoy. A novel framework for cost optimization of renewable energy installations: A case study of nigeria. *Energies*, 12(22):4311, 2019.

[9] Christoph Bartneck, Tony Belpaeme, Friederike Eyssel, Takayuki Kanda, Merel Keijsers, and Selma Šabanović. *Human-robot interaction: An introduction*. Cambridge University Press, 2020.

[10] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, 35:268–308, 01 2001.

[11] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl_3):7280–7287, 2002.

[12] Michael Brand, Michael Masuda, Nicole Wehner, and Xiao-Hua Yu. Ant colony optimization algorithm for robot path planning. In *2010 International Conference On Computer Design and Applications*, volume 3, pages V3–436–V3–440, Qinhuangdao, China, 2010. IEEE.

[13] Cristiano Castelfranchi and Rino Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *Proceedings International Conference on Multi Agent Systems (Cat. No. 98EX160)*, pages 72–79. IEEE, 1998.

[14] Claudia Cavallaro, Carolina Crespi, Vincenzo Cutello, Mario Pavone, and Francesco Zito. Group dynamics in memory-enhanced ant colonies: The influence of colony division on a maze navigation problem. *Algorithms*, 17(2):63, 2024.

[15] Jin-Hee Cho, Kevin Chan, and Sibel Adali. A survey on trust modeling. *ACM Computing Surveys (CSUR)*, 48(2):1–40, 2015.

[16] Chang-Chi Chou, Wen-Chu Chiang, and Albert Y Chen. Emergency medical response in mass casualty incidents considering the traffic

congestions in proximity on-site and hospital delays. *Transportation research part E: logistics and transportation review*, 158:102591, 2022.

[17] M.J. North C.M. Macal. *Tutorial on agent-based modelling and simulation*, 2010.

[18] Piero Consoli, Alessio Collera, and Mario Pavone. Swarm intelligence heuristics for graph coloring problem. In *2013 IEEE Congress on Evolutionary Computation*, pages 1909–1916, Cancun, Mexico, 2013. IEEE, IEEE.

[19] Carolina Crespi, Georgia Fargetta, Mario Pavone, and Rocco A Scollo. An agent-based model for crowd simulation. In *Italian Workshop on Artificial Life and Evolutionary Computation*, pages 15–26. Springer, 2022.

[20] Carolina Crespi, Georgia Fargetta, Mario Pavone, and Rocco A Scollo. An agent-based model to investigate different behaviours in a crowd simulation. In *Bioinspired Optimization Methods and Their Applications: 10th International Conference, BIOMA 2022, Maribor, Slovenia, November 17–18, 2022, Proceedings*, pages 1–14. Springer, 2022.

[21] Carolina Crespi, Georgia Fargetta, Mario Pavone, Rocco A. Scollo, and Laura Scrimali. A game theory approach for crowd evacuation modelling. In Bogdan Filipič, Edmondo Minisci, and Massimiliano Vasile, editors, *Bioinspired Optimization Methods and Their Applications (BIOMA2020)*, volume 12438 of *Lecture Notes in Computer Science*, pages 228–239, Cham, 2020. Springer.

[22] Carolina Crespi and Mario Pavone. Does a group's size affect the behavior of a crowd? Springer, 2024.

[23] Carolina Crespi, Rocco A. Scollo, Georgia Fargetta, and Mario Pavone. How a different ant behavior affects on the performance of the whole colony. In Luca Di Gaspero, Paola Festa, Amir Nakib, and Mario Pavone, editors, *Metaheuristics*, pages 187–199, Cham, 2023. Springer International Publishing.

[24] Carolina Crespi, Rocco A Scollo, Georgia Fargetta, and Mario Pavone. A sensitivity analysis of parameters in an agent-based model for crowd simulations. *Applied Soft Computing*, 146:110684, 2023.

[25] Carolina Crespi, Rocco A. Scollo, and Mario Pavone. Effects of different dynamics in an ant colony optimization algorithm. In *2020 7th International Conference on Soft Computing Machine Intelligence (ISCMI2020)*, pages 8–11. IEEE, 11 2020.

[26] C. Mattiussi D. Floreano. Companion slides for the book bio-inspired artificial intelligence: Theories, methods, and technologies, 2008.

[27] Wu Deng, Junjie Xu, and Huimin Zhao. An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. *IEEE Access*, 7:20281–20292, 2019.

[28] M. Dorigo and G. Di Caro. Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC99)*, volume 2, pages 1470–1477, Washington, DC, USA, 1999. IEEE.

[29] P. Erdös and A. Rényi. On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290, 1959.

[30] Stefka Fidanova and Petrică Pop. An improved hybrid ant-local search algorithm for the partition graph coloring problem. *Journal of Computational and Applied Mathematics*, 293:55–61, 2016.

[31] Riccardo Gervasi, Federico Barravecchia, Luca Mastrogiacomo, and Fiorenzo Franceschini. Applications of affective computing in human-robot interaction: State-of-art and challenges for manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 237(6-7):815–832, 2023.

[32] Melissa Gillis, Ryley Urban, Ahmed Saif, Noreen Kamal, and Matthew Murphy. A simulation–optimization framework for optimizing response strategies to epidemics. *Operations Research Perspectives*, 8:100210, 2021.

[33] Chi Keong Goh, Kay Chen Tan, DS Liu, and Swee Chiang Chiam. A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design. *European Journal of Operational Research*, 202(1):42–54, 2010.

[34] M.S. Granovetter. The Strength of Weak Ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

[35] Shenkai Gu, Ran Cheng, and Yaochu Jin. Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing*, 22:811–822, 2018.

[36] Iman Habibi, Effat S Emamian, and Ali Abdi. Quantitative analysis of intracellular communication and signaling errors in signaling networks. *BMC systems biology*, 8:1–16, 2014.

[37] Ivana Hartmann Tolić, Emmanuel Karlo Nyarko, and Avishai Ceder. Optimization of public transport services to minimize passengers' waiting times and maximize vehicles' occupancy ratios. *Electronics*, 9(2):360, 2020.

[38] Jürgen Heinze. Conflict and conflict resolution in social insects. In *Animal behaviour: Evolution and mechanisms*, pages 151–178. Springer, 2010.

[39] Zhi-Min Huang, Wei-Neng Chen, Qing Li, Xiao-Nan Luo, Hua-Qiang Yuan, and Jun Zhang. Ant colony evacuation planner: An ant colony system with incremental flow assignment for multipath crowd evacuation. *IEEE Transactions on Cybernetics*, 51(11):5559–5572, 2021.

[40] M Janga Reddy and D Nagesh Kumar. Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review. *h2oj*, 3(1):135–188, 2020.

[41] Michael D Jennions and David W Macdonald. Cooperative breeding in mammals. *Trends in Ecology & Evolution*, 9(3):89–93, 1994.

[42] Ya-Hui Jia, Yi Mei, and Mengjie Zhang. A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem. *IEEE Transactions on Cybernetics*, 52(10):10855–10868, 2021.

[43] Raka Jovanovic, Milan Tuba, and Stefan Voß. An efficient ant colony optimization algorithm for the blocks relocation problem. *European Journal of Operational Research*, 274(1):78–90, 2019.

[44] Abdulğaffar Kaya, Pete Bettinger, Kevin Boston, Ramazan Akbulut, Zennure Ucar, Jacek Siry, Krista Merry, and Chris Cieszewski. Optimisation in forest management. *Current Forestry Reports*, 2:1–17, 2016.

[45] Zahra Rezaei Khavas, S Reza Ahmadzadeh, and Paul Robinette. Modeling trust in human-robot interaction: A survey. In *International conference on social robotics*, pages 529–541. Springer, 2020.

[46] Gerta Köster, Michael Seitz, Franz Treml, Dirk Hartmann, and Wolfram Klein. On modelling the influence of group formations in a crowd. *Contemporary Social Science*, 6(3):397–414, 2011.

[47] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.

[48] Chaochao Li, Pei Lv, Dinesh Manocha, Hua Wang, Yafei Li, Bing Zhou, and Mingliang Xu. Acsee: Antagonistic crowd simulation model with emotional contagion and evolutionary game theory. *IEEE transactions on affective computing*, 13(2):729–745, 2019.

[49] Ziwei Li, Huang Huang, Nan Li, Mei Ling Chu Zan, and Kincho Law. An agent-based simulator for indoor crowd evacuation considering fire impacts. *Automation in Construction*, 120:103395, 2020.

[50] Boliang Lin, Yinan Zhao, and Ruixi Lin. Optimization for courier delivery service network design based on frequency delay. *Computers & Industrial Engineering*, 139:106144, 2020.

[51] Hong Liu, Bin Xu, Dianjie Lu, and Guijuan Zhang. A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm. *Applied Soft Computing*, 68:360–376, 2018.

[52] Ilias Lymperopoulos and George Lekakos. Analysis of social network dynamics with models from the theory of complex adaptive systems. In Christos Douligeris, Nineta Polemi, Athanasios Karantjias, and Winfried Lamersdorf, editors, *Collaborative, Trusted and Privacy-Aware e/m-Services*, pages 124–140, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[53] Charles M Macal and Michael J North. Tutorial on agent-based modeling and simulation. In *Proceedings of the Winter Simulation Conference, 2005.*, pages 14–pp, Orlando, FL, USA, 2005. IEEE, IEEE.

[54] Franjo Matkovic, Marina Ivasic-Kos, and Slobodan Ribaric. A new approach to dominant motion pattern recognition at the macroscopic crowd level. *Engineering Applications of Artificial Intelligence*, 116:105387, 2022.

[55] Inaki Maurtua, Aitor Ibarguren, Johan Kildal, Loreto Susperregi, and Basilio Sierra. Human–robot collaboration in industrial applications: Safety, interaction and trust. *International Journal of Advanced Robotic Systems*, 14(4):1729881417716010, 2017.

[56] Azade Mehri, Abdolrassoul Salmanmahiny, Seyed Hamed Mirkarimi, and Hamid Reza Rezaei. Use of optimization algorithms to prioritize protected areas in mazandaran province of iran. *Journal for Nature Conservation*, 22(5):462–470, 2014.

[57] Katherine L Milkman, Laura Huang, and Maurice E Schweitzer. Teetering between cooperation and competition: How subtle cues unexpectedly derail coopetitive workplace relationships. Technical report, Citeseer, 2014.

[58] Mark Newman. *Networks*. Oxford university press, 2018.

[59] C Thi Nguyen. Competition as cooperation. *Journal of the Philosophy of Sport*, 44(1):123–137, 2017.

[60] Lei Pan, Huan Zhou, Yun Liu, and Minghui Wang. Global event influence model: integrating crowd motion and social psychology for global anomaly detection in dense crowds. *Journal of Electronic Imaging*, 28(2):023033–023033, 2019.

[61] Xiaoshan Pan, Charles S. Han, Ken Dauber, and Kincho H. Law. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *AI & SOCIETY*, 22(2):113–132, 11 2007.

[62] Huijun Peng, Chun Ying, Shuhua Tan, Bing Hu, and Zhixin Sun. An improved feature selection algorithm based on ant colony optimization. *IEEE Access*, 6:69203–69209, 2018.

[63] Camelia-M. Pintea, Oliviu Matei, Rabie A. Ramadan, Mario Pavone, Muaz Niazi, and Ahmad Taher Azar. A fuzzy approach of sensitivity for multiple colonies on ant colony optimization. In Valentina Emilia Balas, Lakhmi C. Jain, and Marius Mircea Balas, editors, *Soft Computing Applications*, pages 87–95, Cham, 2018. Springer International Publishing.

[64] Sarvapali D Ramchurn, Dong Huynh, and Nicholas R Jennings. Trust in multi-agent systems. *The knowledge engineering review*, 19(1):1–25, 2004.

[65] Paul Robinette, Ayanna M Howard, and Alan R Wagner. Effect of robot performance on human–robot trust in time-critical situations. *IEEE Transactions on Human-Machine Systems*, 47(4):425–436, 2017.

[66] Paul Robinette, Wenchen Li, Robert Allen, Ayanna M Howard, and Alan R Wagner. Overtrust of robots in emergency evacuation scenarios. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 101–108. IEEE, 2016.

[67] António MM Rodrigues, Jessica L Barker, and Elva JH Robinson. From inter-group conflict to inter-group cooperation: insights

from social insects. *Philosophical Transactions of the Royal Society B*, 377(1851):20210466, 2022.

[68] Marta Romeo, Peter E McKenna, David A Robb, Gnanathusharan Rajendran, Birthe Nesset, Angelo Cangelosi, and Helen Hastie. Exploring theory of mind for human-robot collaboration. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 461–468. IEEE, 2022.

[69] Mahmoud Saleh, Yusef Esa, and Ahmed Mohamed. Applications of complex network analysis in electric power systems. *Energies*, 11:1381, 05 2018.

[70] Mahmoud Saleh, Yusef Esa, Nwabueze Onuorah, and Ahmed A Mohamed. Optimal microgrids placement in electric distribution systems using complex network framework. In *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 1036–1040, San Diego, CA, 2017. IEEE, IEEE.

[71] Hamizan Sharbini, Roselina Sallehuddin, and Habibollah Haron. Crowd evacuation simulation model with soft computing optimization techniques: a systematic literature review. *Journal of Management Analytics*, 8(3):443–485, 2021.

[72] Nur Siyam, Omar Alqaryouti, and Sherief Abdallah. Research issues in agent-based simulation for pedestrians evacuation. *IEEE Access*, 8:134435–134455, 2020.

[73] Iryna Sotnyk, Tetiana Kurbatova, Yaroslavna Romaniuk, Olha Prokopenko, Viktoriya Gonchar, Yuriy Sayenko, Gunnar Prause, and Aleksander Sapiński. Determining the optimal directions of investment in regional renewable energy development. *Energies*, 15(10):3646, 2022.

[74] J.P. Steiner. Maze maker, 2004.

[75] Paula Stockley and Jakob Bro-Jørgensen. Female competition and its evolutionary consequences in mammals. *Biological Reviews*, 86(2):341–366, 2011.

[76] E.G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, Hoboken, New Jersey, 2009.

[77] Dedra Townsend and AmirHossein MajidiRad. Trust in human-robot interaction within healthcare services: A review study. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 86281, page V007T07A030. American Society of Mechanical Engineers, 2022.

[78] Ashutosh Trivedi and Shrisha Rao. Agent-based modeling of emergency evacuations considering human panic behavior. *IEEE Transactions on Computational Social Systems*, 5(1):277–288, 2018.

[79] Arthur Turrell. Agent-based models: understanding the economy from the bottom up. *Bank of England Quarterly Bulletin*, 2016 Q4:Q4, 2016.

[80] C. Natalie van der Wal, Daniel Formolo, Mark A. Robinson, Michael Minkov, and Tibor Bosse. *Simulating Crowd Evacuation with Socio-Cultural, Cognitive, and Emotional Elements*, pages 139–177. Springer International Publishing, Cham, 2017.

[81] Wouter van Toll, Fabien Grzeskowiak, Axel López Gandía, Javad Amirian, Florian Berton, Julien Bruneau, Beatriz Cabrero Daniel, Alberto Jovane, and Julien Pettré. Generalized microscropic crowd simulation using costs in velocity space. In *Symposium on Interactive 3D Graphics and Games*, I3D '20, New York, NY, USA, 2020. Association for Computing Machinery.

[82] Danielle M Varda, Rich Forgette, David Banks, and Noshir Contractor. Social network methodology in the study of disasters: Issues and insights prompted by post-katrina research. *Population research and policy review*, 28:11–29, 2009.

[83] Michela Vezzoli and Cristina Zogmaister. An introductory guide for conducting psychological research with big data. *Psychological Methods*, 28:580–599, 2023.

[84] Alan R Wagner and Ronald C Arkin. Recognizing situations that demand trust. In *2011 RO-MAN*, pages 7–14. IEEE, 2011.

[85] Qingqing Wang, Hong Liu, Kaizhou Gao, and Le Zhang. Improved multi-agent reinforcement learning for path planning-based crowd simulation. *IEEE Access*, 7:73841–73855, 2019.

[86] Shouna Wang, Hong Liu, Kaizhou Gao, and Jianxin Zhang. A multi-species artificial bee colony algorithm and its application for crowd simulation. *IEEE Access*, 7:2549–2558, 2019.

[87] Thomas Weise. Measuring the runtime of (optimization) algorithms, 2017.

[88] Qianjiao Wu, Rong Lan, and Wei Zhou. Traffic flow simulation based on adaptive agent. In *2018 3rd International Conference on Smart City and Systems Engineering (ICSCSE)*, pages 697–700, Xiamen, China, 2018. IEEE, IEEE.

[89] Wei Xie, Eric Wai Ming Lee, Tao Li, Meng Shi, Ruifeng Cao, and Yuchun Zhang. A study of group effects in pedestrian crowd evacuation: Experiments, modelling and simulation. *Safety Science*, 133:105029, 2021.

[90] Bo Xing, Wen-Jing Gao, Bo Xing, and Wen-Jing Gao. Imperialist competitive algorithm. *Innovative computational intelligence: A rough guide to 134 clever algorithms*, pages 203–209, 2014.

[91] Yanchun Xu, Haiquan Liu, Shasha Xie, Lei Xi, and Mi Lu. Competitive search algorithm: a new method for stochastic optimization. *Applied Intelligence*, 52(11):12131–12154, 2022.

[92] G. Yang, Y. Chen, and J.P. Huang. The highly intelligent virtual agents for modeling financial markets. *Physica A: Statistical Mechanics and its Applications*, 443:98 – 108, 2016.

[93] Shanwen Yang, Tianrui Li, Xun Gong, Bo Peng, and Jie Hu. A review on crowd simulation and modeling. *Graphical Models*, 111:101081, 2020.

[94] SONG YANG. Networks: An introduction by m. e. j. newman. *The Journal of Mathematical Sociology*, 37(4):250–251, 2013.

[95] Xiaoxia Yang and Qianling Wang. Crowd hybrid model for pedestrian dynamic prediction in a corridor. *IEEE Access*, 7:95264–95273, 2019.

[96] Sangseok You and Lionel Robert. Trusting robots in teams: Examining the impacts of trusting robots on team performance and satisfaction. In *You, S. and Robert, LP (2019). Trusting robots in teams: Examining the impacts of trusting robots on team performance and satisfaction, proceedings of the 52th hawaii international conference on system sciences, Jan*, pages 8–11, 2018.

[97] Sangseok You and Lionel P Robert Jr. Team robot identification theory (trit): robot attractiveness and team identification on performance and viability in human–robot teams. *The Journal of Supercomputing*, 78(18):19684–19706, 2022.

[98] Yufei Yuan, Bernat Goñi-Ros, Ha H Bui, Winnie Daamen, Hai L Vu, and Serge P Hoogendoorn. Macroscopic pedestrian flow simulation using smoothed particle hydrodynamics (sph). *Transportation research part C: emerging technologies*, 111:334–351, 2020.

[99] Furkan Yücel and Elif Sürer. Implementation of a generic framework on crowd simulation: a new environment to model crowd behavior and design video games. *Mugla Journal of Science and Technology*, 6:69–78, 2020.

[100] Dehui Zhang, Xiaoming You, Sheng Liu, and Han Pan. Dynamic multi-role adaptive collaborative ant colony optimization for robot path planning. *IEEE Access*, 8:129958–129974, 2020.

[101] Xiaoping Zheng and Yuan Cheng. Modeling cooperative and competitive behaviors in emergency evacuation: A game-theoretical approach. *Computers & Mathematics with Applications*, 62(12):4627–4634, 2011.

[102] Zi-Xuan Zhou, Wataru Nakanishi, and Yasuo Asakura. Route choice in the pedestrian evacuation: Microscopic formulation based on visual information. *Physica A: Statistical Mechanics and its Applications*, 562:125313, 2021.

[103] Xinlu Zong, Jingxi Yi, Chunzhi Wang, Zhiwei Ye, and Naixue Xiong. An artificial fish swarm scheme based on heterogeneous pheromone for emergency evacuation in social networks. *Electronics*, 11(4):649, 2022.