Doctoral Dissertation

Doctoral Program in Electrical, Electronics and Communications Engineering

(36^{th}cycle)

Human-centered robotics: challenges in socially-aware navigation and mission planning

Giada Galati

******

Supervisor:

Prof. Alessandro Rizzo

Doctoral Examination Committee:

Prof. Mattia Frasca, Università di Catania, Italy

Prof. Marina Indri, Politecnico di Torino, Italy

Prof. David Naso, Referee, Politecnico di Bari, Italy

Prof. Domenico Prattichizzo, Referee, Università di Siena

Prof. Kimon Valavanis, University of Denver, USA

Politecnico di Torino

2024

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Giada Galati
2024

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

To my loving parents and my life partner, Andrea.
To failure and perseverance.
To the management of the emotional roller coaster of research.

Acknowledgements

Ogni volta che concludo un percorso impegnativo, mi chiedo sempre come sia riuscita a raggiungere un traguardo così ambizioso. Mi chiedo quali siano stati i fattori che hanno reso possibile questo successo e sulle differenze rispetto alle mie esperienze precedenti.

Fortuna? Talento? Costanza e perseveranza? Ho provato a capire se ci fosse qualche fattore che ha influenzato di più il percorso oppure se esista un connubio tra fattori che ha permesso tutto questo. Penso che la fortuna sia troppo imprevedibile e da sola sicuramente non sarebbe bastata per completare un dottorato. Penso che sia lo stesso anche per il talento. Quest'ultimo sicuramente può essere una buona base per creare e ottenere qualcosa di ambizioso, ma ovviamente non basta. Cosa dire della costanza e della perseveranza? Ritengo siano la chiave di ogni successo ma non l'unica. Penso che senza il duro lavoro quotidiano non si possa raggiungere nessun risultato di qualità. Tuttavia, intraprendere un dottorato significa immergersi in un'altalena di emozioni: l'ebbrezza della scoperta si scontra con la fatica della ricerca, la soddisfazione dei progressi si accompagna all'insoddisfazione per le battute d'arresto, il tutto in un arco temporale di tre anni. In queste situazioni se si ha soltanto la costanza e la perseveranza non si raggiunge facilmente l'obiettivo. Quindi penso ci sia un ulteriore fattore da considerare oltre quelli sopra citati: la gestione delle emozioni. Infatti, se emotivamente non si è pronti ad affrontare una situazione, anche se si hanno la costanza e la voglia di fare, purtroppo non si va da nessuna parte. Quindi probabilmente la mia visione di successo non è altro che un connubio perfetto tra disciplina, capacità intellettuali ed emotive.

Tuttavia, il quadro per spiegare il fenomeno non è ancora completo, sicuramente un ruolo fondamentale lo hanno avuto tutte le persone che hanno creduto in me e che mi hanno sostenuto in questi anni.

In primis vorrei ringraziare il mio supervisor, il Professor Rizzo, per essere stato sempre molto disponibile e per avermi dato la possibilità di lavorare con un vero robot e di collaborare con Leonardo Labs. Ringrazio il Professor Simone Macrì, dell'Istituto Superiore di Sanità, per avermi aiutato ad avere una prospettiva interdisciplinare sui progetti presentati in questa tesi. Ringrazio Stefano per aver condiviso con me preziosi consigli riguardo l'insegnamento in università. Ringrazio David, Lorenzo, Weibin and Yuan per essere stati fondamentali nel reclutamento dei partecipanti per l'esperimento con il robot. A Mattia, compagno di ufficio e di dottorato su cui ho sempre potuto contare per un consiglio, una pizza o solo per una chiacchierata. Ad Andrea e Giacomo, collaboratori fenomenali! Grazie a voi sono riuscita a comprendere quanto un team affiatato possa fare la differenza sul lavoro. Grazie.

Inoltre, ringrazio tutto il dipartimento e i dottorandi del DET per rendere il laboratorio un luogo felice e confortevole per affrontare le sfide quotidiane.

Ma il ringraziamento più grande di tutti va ai miei genitori. Loro hanno sempre creduto in me, anche quando le mie scelte accademiche si discostavano dai loro standard. Non comprendevano a pieno il mio percorso, ma nutrivano e nutrono una grandissima fiducia in me. Hanno sempre puntato su di me, concedendomi la libertà di scegliere, e credo che questo sia stato fondamentale per far emergere il meglio di me. Li ringrazio per il sostegno morale che mi hanno offerto e per la forza dimostrata nel rinunciare alla mia presenza quotidiana in nome del mio successo e della mia felicità.

Un ringraziamento speciale va a Peppe, che è saputo entrare nella mia vita con delicatezza, dimostrandomi affetto e premura in ogni momento. Grazie anche per i pacchi da giù sempre molto graditi!

Un sentito grazie va a Simona, per le lunghe chiamate del weekend in cui ci si raccontava tutti gli aggiornamenti più importanti. Grazie per il supporto, i preziosi consigli e per i viaggetti organizzati all'ultimo momento!

Grazie a Chiara T., regina del riconoscimento e della gestione delle emozioni. Grazie per avermi donato una tavolozza piena di colori con cui interpretare la vita.

A Giulia, Chiara e Francesco, amici storici su cui si può sempre contare.

Al mio compagno di vita, Andrea. Grazie per essere stato quotidianamente presente durante questo percorso nonostante a volte si presentasse come una giostra

di emozioni difficile da gestire. Grazie per avermi sempre supportato nelle mie scelte e ascoltato pazientemente. Grazie per tutto l'amore che mi hai dato e mi darai. Sei casa.

Abstract

Over the past decade, remarkable advances in science, engineering, and technology have catalyzed the expansion of robotics. This transformative progress has given rise to a surge in the widespread integration of robots that interact directly with humans. In the scope of this dissertation, emphasis is placed on the examination of human interaction with robots. Specifically, the study delves into two distinct categories of interaction: firstly, the *bystander*, i.e. a human who assumes a non-interactive role with the robot but coexists in the same physical space, and secondly, the *supervisor*, i.e. a human who oversees robots and makes decisions related to critical situations.

Some examples of *bystander-robot* interaction are the offices, hospital and hotel. In such context, the main aim of the human is the avoidance, instead, the main aim of the robot is to guarantee not only physical safety but also consider the psychological safety associated with coexisting in an environment with humans. To achieve an optimal integration of both types of safety, it is necessary that robot moves intending to minimize any disturbance to pedestrians, thus seeking to ensure its social acceptability. This goal is achieved through the concept of socially-aware navigation. Thus, in this thesis *two* socially-aware navigation algorithms are developed: the first is based on game-theory (GT), and the second is based on the social force model and the game theory (GTSFM). Each algorithm underwent rigorous evaluation, using state-of-the-art *quantitative* metrics and collecting *qualitative* information through participant-administered questionnaires.

Compared to the current state-of-the-art method, the GT algorithm achieved better performance according to both types of evaluation.

Regarding the *quantitative* evaluation, GTSFM exhibited a smoother path compared to the two state-of-the-art algorithms, resulting in a more natural motion. The *qualitative* analysis, performed with a real-world experiment, did not identify any algorithm that showed significant superiority over the others. This lack of distinction

can be attributed to unaccounted factors. The robot's appearance could have obscured the distinction between the algorithms. Additionally, the limited velocity of the real robot may have limited the range of conditions tested among the different algorithms and, consequently, obscured the distinction between them.

These results represent a significant milestone in advancing the integration of robots into social environments also leaving important hints for future research development.

On the other hand, some instances of *supervisor-robots* interaction are evident in scenarios characterized by complex and critical tasks such as rescue operations. In such context, a multi-robot autonomous system is involved to handle the mission. Instead, the human is responsible for crucial decision-making, particularly assessing the mission's overall success, ensuring the safety of human lives, and managing unforeseen situations. Within this context, coordinating a multi-robot system with human supervision is crucial. Thus, in this dissertation, a task allocation in a dynamic environment is proposed obtaining promising results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past decade, advances in science, engineering, and technology have influenced the growth of robotics, creating a notable escalation in the proliferation of robots engaging directly with humans [8].

Historically, the utilization of robots was primarily confined to industrial contexts, due to their adeptness in executing repetitive tasks to increase productivity and quality of the work [9].

Currently, the growing accessibility and competitive pricing of service robots are driving their integration into homes, potentially impacting daily life. The growing prevalence of service robots is not just a possibility but a tangible reality, with examples of mobile robots coexisting harmoniously with humans. Notably, the strides made in healthcare service robotics, catalyzed by the challenges presented by COVID-19, highlight the importance and the substantial impact that these technological advancements can have on our well-being [10]. Moreover, the spectrum of applications for robots has expanded across different domains such as: space, agriculture, construction, hazardous scenarios, surveillance and security, to name a few [11].

The trends and future directions in robotics strongly indicate that the pervasive utilization of robots is ready to revolutionize numerous facets of our world, surpassing the bounds of imagination [8].

Depending on the application, humans have the opportunity to engage with robots through various modes of interaction. At the state of the art [12], five distinct human

roles are defined: the *supervisor*, the *operator*, the *collaborator*, the *cooperator*, and the *bystander*.

The *supervisor* oversees the robot's operations and provides instructive guidance on task execution. Moreover, in critical scenarios (such as rescue missions), the *supervisor* has a crucial role in decision-making due to ethical issues associated with managing the situation [13].

The *operator* is a human that controls the robot. During this interaction, the human is placed in a higher hierarchical position than the robot, which operates under human control. An example of this interaction is the surgical da Vinci robot [14], which is directly controlled by the surgeon ensuring precise movements and eliminating the possible hand tremor of the surgeon.

The *collaborator* shares the goal with the robot, forming a symbiotic relationship in which mutual success is based on cooperation. In this case, the human collaborator, although dependent on the actions of the robot, is not burdened with managerial duties. An example of this interaction is an industrial setting where a robot assists a human counterpart by securely holding and rotating hefty workpieces. This collaborative interaction emphasizes a lack of hierarchical distinctions between the human and the robot, fostering an environment of equality and shared responsibility for task accomplishment [12].

Also the *cooperator* interacts with the robot to achieve a common goal. Unlike the *collaborator* role, the *cooperator* is not directly dependent on the robot's actions due to a strict task allocation between the human and the robot. Thus, both partners contribute to the completion of the task independently, but their combined efforts are essential for achieving the shared overall goal. This cooperative scenario is used in the pick-and-place robots in manufacturing. Here, the human and the robot are co-workers on an assembly line, each undertaking specific tasks autonomously but collectively working towards the shared objective of producing a certain product [12].

The *bystander* is a human who assumes a non-interactive role with the robot but shares the same physical space. Despite lacking direct interaction, the human must maintain a cognitive awareness of the robot and its movements to prevent collisions. Some examples of such human-robot interaction are the use of mobile robots in museums [15], offices [16], hospitals [17], and hotels [18]. In all these scenarios, even in the absence of shared goals, humans and robots must synchronize their actions to effectively navigate in the shared space and prevent conflicts.

Fig. 1.1 Role of humans as *bystanders* in a human-robot shared environment.

This thesis focuses on two distinct human roles, namely the *bystander* and the *supervisor*. Firstly, this thesis explores the role of the human as a *bystander*, studying the design of some trajectory planning to enhance human-robot interaction (see Figure 1.1). Secondly, this thesis examines the role of the human as a *supervisor* in a critical scenario involving the coordination of a multi-robot system to accomplish a shared objective (see Figure 1.2).

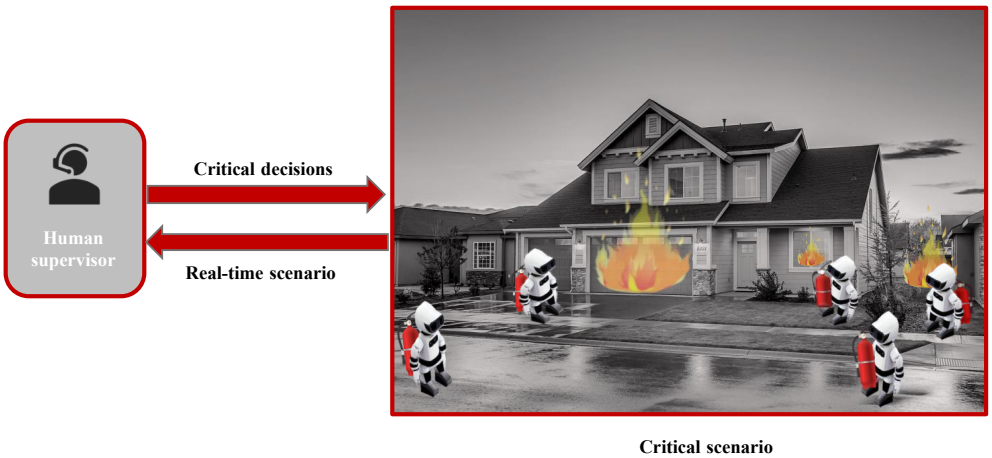


Fig. 1.2 Role of human as *supervisor* within a multi-robots system in a critical scenario.

1.1 Thesis contribution in socially-aware navigation context

In the *bystander-robot* interaction, the aim of the human is the avoidance, while the robot aims to navigate respecting both the physical and psychological safety of humans sharing the environment. To achieve optimal integration of both safety aspects, robots must prioritize pedestrian comfort by minimizing disturbance during their movement towards the goal, thereby ensuring its social acceptability. This objective is realized through the concept of socially-aware navigation, which is defined in [19] as:

"the strategy exhibited by a social robot which identifies and follows social conventions (in terms of management of space) in order to preserve a comfortable interaction with humans. The resulting behavior is predictable, adaptable and easily understood by humans."

Thus, this thesis aims to provide steps forward to design a socially-aware motion planning for an autonomous robot to be acceptable to humans and not annoy them.

Considering this ambitious goal, navigation is transformed from a purely technical task to a complex interdisciplinary challenge, combining perception, dynamical system theory, social conventions, human motion modeling, and psychology.

In the pursuit of enhancing trust, comfort, and social acceptance, robots must recognize humans as intelligent agents capable of interacting and influencing each other's movements. Current approaches in socially-aware navigation often oversimplify human motion models by portraying humans as dynamic obstacles or employing basic motion models [20, 21]. These oversimplifications fail to describe the intricate nature of the interactions between humans, resulting in trajectories that lack predictability, smoothness, and, consequently, fail to meet human acceptance standards. Learning models [22–25] offer a promising alternative, with the potential for superior outcomes. However, this approach relies heavily on access to extensive training data involving human subjects, a resource that is not always readily available.

In this thesis, we present *two* socially-aware robot navigation algorithms that accurately model human behavior using game theory. Game theory stands out as a powerful choice, presenting considerable advantages over alternative model-

ing approaches like reactive strategies [20, 21] and learning schemes [22–25]. In contrast to reactive strategies, game theory excels in its capability to predict and anticipate motion, modelling the decision-making inherent in human social contexts. In comparison to learning schemes, it surpasses challenges related to explainability, generalization, and the often demanding requirement for extensive training datasets.

While ensuring the social acceptability of algorithms remains challenging due to the inherent subjectivity and context-dependence of the concept, Kruse et al. [26] propose a study that identifies three key approaches to enhancing robot acceptance: human comfort, naturalness, and sociability. Then, building upon the classification framework presented in [26], the authors in [27] assign to each approach the most suitable measurement method, as described below.

(i) *Human comfort* involves ensuring an interaction between humans and robots avoiding any annoyance or stress. Human comfort comprises both physical and psychological dimensions, assessed through objective spatial models and subjective participant evaluations, respectively. Regarding the former, studies demonstrate a well-established link between interpersonal distance and human comfort levels [28]. In particular, one of the key factors influencing pedestrian comfort during robot interaction is the robot's ability to respect individual *personal space* [29]. Regarding psychological comfort, the state of the art provides different types of questionnaires such as the Godspeed questionnaire [30], the BEHAVE-II instrument [31], and the Human-Robot Interaction Evaluation Scale (HRIES) [7];

(ii) *Naturalness* is the likeness of robots to humans in low-level behavior patterns. Specifically, the naturalness is assessed by considering the *human-likeness* and the *smoothness* of robot movements [27], considering both geometric and velocity profile aspects. Beyond quantitative approaches, qualitative scales can assess naturalness of movement by evaluating the animacy of robotic behavior [7];

(iii) *Sociability* refers to how well the robot interacts with humans in a socially acceptable manner. In this regard, the HRIES scale also includes a *sociability* sub-dimensions, which *qualitatively* assesses this factor through a questionnaire. Moreover, according to [27], sociability refers to the compliance with explicit cultural social conventions in navigation behavior. Examples of such behavior include walking on the right side of a hallway or standing in a queue. These behaviors typically reflect socially imposed constraints and can contribute to conflict resolution and establishing social order in navigation. However, ensuring sociability is highly

complex, influenced by both the target culture and the specific context in which the robot is deployed. For these reasons, to assess *quantitatively* the robot movement, this thesis employs the first two approaches, omitting the utilization of the last one. This choice ensures the development of generalizable algorithms that do not conform to specific social cultures. Notably, both algorithms developed in this thesis are designed to ensure physical *comfort* through the respect of human personal space. Regarding the *naturalness*, the first algorithm exploits the concept of *human-likeness*. For the second algorithm instead, naturalness is quantitatively evaluated by analyzing the *smoothness* of planned trajectories through performance metrics of the state of the art.

Moreover, both algorithms are *qualitative* evaluated through questionnaires, administered to volunteers to comprehend how humans perceive trajectories.

During the *qualitative* evaluation, both algorithms exploit the concept of anthropomorphism, which reflects humans' innate tendency to attribute intentions and consciousness to non-human entities [32]. As a result of this attribution, the development of robotic trajectories sharing certain aspects of human movement, strengthens anthropomorphism and increases acceptability among humans [33].

The first algorithm is *qualitatively* evaluated through a variation of the Turing test administered in the form of an online survey questionnaire to a pool of 691 participants. The anthropomorphism is evaluated by exploiting the *human-likeness* measurement. Results reveal that although the first algorithm achieves superior performance to the state-of-the-art, the participants found little difference between the trajectories generated by our algorithm and those executed by humans.

On the other hand, the *qualitative* analysis of the second algorithm is performed through an experiment with a real robot and a pool of 76 people. To verify anthropomorphism qualitatively, the HRIES scale is used, which includes measures regarding *comfort*, *naturalness* and *sociability*. The results show that the participants failed to identify a single algorithm that demonstrated significant superiority over the others. This absence of differentiation could be attributed to unaccounted factors. The robot's appearance could have obscured the distinction between the algorithms. Additionally, the limited velocity of the real robot may have limited the range of conditions tested among the different algorithms and, consequently, obscured the distinction between the algorithms.

These findings represent a significant step forward in integrating robots into social environments and provide valuable insights for future research advancements.

1.2 Thesis contribution in Multi-robot system context

The increasing complexity of tasks in various domains, coupled with the limitations of single robots, has fueled the development of multi-robot systems (MRS). These systems, comprising multiple robots collaborating to achieve a common goal.

Unlike traditional single-robot systems, MRS leverages the power of collective intelligence and distributed capabilities, opening up new horizons in various domains, including rescue operation [34], large scale assembly [35], hazardous waste cleanup [36], security [37], and agriculture [38].

The key motivation behind MRS lies in the recognition that a group of robots working together can outperform individual units in terms of efficiency, flexibility, and adaptability. The potential benefits extend to enhanced productivity, robustness, and scalability, making MRS an increasingly attractive solution for addressing real-world problems.

Complex and life-critical tasks, like rescue operations, frequently necessitate the utilization of MRS involving unmanned vehicles and human operator. This strategic combination ensures a comprehensive approach, leveraging the strengths of both unmanned technologies and human expertise to effectively address the complexities inherent in such critical tasks. While advanced robotic technologies provide their capabilities in performing tasks in hazardous environments, a human *supervisor* provides indispensable qualities such as adaptability, intuition, decision-making and responsibility that are challenging for autonomous systems to replicate entirely. Furthermore, human supervisor possesses the capacity for empathy and ethical considerations, which are crucial in situations where delicate choices have to be made. The supervision ensures perfect integration with the robotic team, allowing real-time adjustments and strategy refinement. This calls for the design of robust and efficient mission planning for MRSs with human supervisor.

This thesis investigates the design of a mission planning system for MRS working alongside a human supervisor in critical situations. The developed approach aims to optimize teamwork through effective MRS coordination, ultimately leading to

mission accomplishment. Notably, our mission planning framework comprises two key components: task allocation and motion planning, facilitating the efficient calculation of task costs.

The work presented in this thesis introduces numerous innovations and enhancements when compared to the existing state of the art. *First*, we propose an auction-based method as task allocation for a heterogeneous team operating in a dynamic scenario with human supervision. This method handles complex real-world challenges like precedence constraints, rescue priorities, and on-the-go re-allocations–a setting that was not entirely contemplated in the past [39–44]. It also incorporates human intervention by dynamically adjusting task allocation through constraints, ensuring safety and adaptation to unforeseen circumstances. *Second*, we employ a multi-goal motion planner in conjunction with an auction-based allocation strategy, resulting in a system that is not only fast and efficient but also responsive to disturbances. In this context, the multi-goal motion planner optimally exploits the features of the auction by simultaneously computing the cost for each robot to execute a designated task. *Third*, unlike existing work, this system integrates human supervision through dynamic constraints on the auction-based task allocator. This promotes ethical decision-making in demanding environments, aligning robot actions with human values and responsibilities.

An extensive simulation campaign in a rescue scenario validates our approach in dynamic scenarios. We highlight the benefits of the proposed multi-goal strategy by comparing it with single-goal motion planning strategies at the state of the art. Finally, we provide evidence for the system's efficiency by demonstrating the powerful synergistic combination of the auction-based allocation and the multi-goal motion planning approach.

1.3 Outline

This thesis is divided into three distinct parts. The first part, titled *"Socially-aware Navigation"*, presents two novel algorithms developed to enhance the robot's social acceptance. Chapter 2 presents an introduction to the key socially-aware navigation concepts used in this part. Then, in Chapter 3 the game theoretical trajectory planning is presented considering both the *quantitative* and *qualitative* results. In Chapter 4, the game theoretical social force model (GTSFM) for human motion prediction

is presented. Then, in Chapter 5, the model developed in Chapter 4 is used to control a mobile robot's movement. In addition, Chapter 5 presents *quantitative* results to assess the algorithm's performance. On the other hand, in Chapter 6, the experimental details for testing the GTSFM algorithm accompanied by the presentation of *qualitative* results are introduced.

The second part of this thesis, titled *"Mission Planning with Human supervision"*, presents an auction-based task allocation and motion planning for multi-robot systems with human supervision in a critical scenario.

Finally, the third part is devoted to the final conclusions of this thesis.

Part I

Part 1 Socially-aware Navigation

Chapter 2

Key Socially-aware Navigation concepts used in this part

This chapter lays the groundwork for the subsequent chapters about socially-aware navigation. Three key concepts are presented: (i) a foundational overview of socially-aware navigation concepts and the rationale for employing game theory; (ii) a comprehensive review of existing literature on socially-aware navigation algorithms; (iii) a detailed explanation of the methodologies chosen in this thesis to evaluate the designed algorithms.

In particular, the first concept is presented in Section 2.1. The latter provides essential context by introducing the notion of socially-aware navigation. Furthermore, since the subsequent chapters will present two game theory-based algorithms, this section will comprehensively justify the rationale behind our design choice of employing game theory in this domain.

The second concept is presented in Section 2.2. The latter is important to understand how the algorithms, presented in the following chapters, fit into the broader picture of the scientific literature.

The last concept, presented in Section 2.3, aims to explain the metrics used to *quantitatively* measure the trajectories generated by the algorithms presented in the following chapters and, in some cases, also the real trajectories of people.

2.1 Socially-aware Navigation: foundational concepts and motivations for employing game theory

The widespread diffusion of service robots for diverse applications is making autonomous robots more and more pervasive in our lives [45]. In the near future, autonomous robots will likely coexist and share our very space, such as hospitals [17], exhibitions [46], and office buildings [16], to name a few. Application scenarios will be characterized by populated and dynamic environments, where autonomous navigation has to ensure not only the physical safety of human subjects, but also a great degree of *social acceptability* [26]. Trajectory planners at the state of the art mostly aimed at ensuring the former requisite [47–50], while seldom tackling the social acceptability issue. This is primarily due to the fact that most contemporary autonomous navigation algorithms model humans as inanimate dynamic obstacles rather than social entities interacting with each other through complex and strategized patterns [51]. However, the oversimplification of human behavioral traits in the design of navigation algorithms may have severe consequences, such as the emergence of the well-known "freezing robot problem" [52]. The freezing robot problem occurs when the environment exceeds a certain level of complexity and the robot is no longer able to manage it because of some deficiencies in the navigation algorithm, for example in the prediction of the human motion model. This context could lead to conditions in which the robot considers all paths unsafe, so it freezes its motion (or makes unnecessary maneuvers) to avoid collision with humans [52].

This is where socially-aware navigation emerges as a fundamental requirement for the design of social robots. Socially-aware navigation gives robots the ability to become from simple obstacle-avoiding entities into socially aware participants in navigation. Through sensor fusion and accurate human-motion predictions, robots can: (i) perceive the social environment through sensors; (ii) understand and adhere to social norms; (iii) make decisions that are not only collision-free but also socially appropriate, considering factors such as cultural norms, and anticipated human behavior; (iv) plan and execute their movements mimic human behavior to enhance human-robot interaction and social acceptability [33]. Thus, socially-aware navigation is a complex concept that combines perception, dynamical system theory, social conventions, human motion modeling, and psychology.

In this thesis, to design the two socially-aware trajectory planning presented in Chapters 3 and 5, we leverage the concept of anthropomorphism, i.e. the intrinsic tendency of humans to attribute intentions and consciousness to non-human entities [32]. Due to such an attribution, designing robotic trajectories that share some features with human trajectories would reinforce anthropomorphism, enhancing the acceptability by humans [33]. Hence, developing accurate models of human motion and decision-making processes while navigating in populated environments is a problem of paramount importance.

Recent efforts in socially-aware navigation model humans as static entities [53] or as agents driven by very simplistic motion models [54]. Such simplistic assumptions may hardly cope with the complexity of human behavior and interaction, yielding trajectories that are far from predictable, smooth, and in turn acceptable for humans. Models based on learning theory, on the other hand, promise better results [55] provided that a large training data set involving human subjects is available, which is not always the case.

Game theory is a powerful and flexible mathematical paradigm to study strategic interactions between different components of a system (in our scenario, moving individuals) [56]. Because of such flexibility, game theory has rapidly increased in popularity over the past decades and has found application in interdisciplinary studies in multi-agent interactions in many fields, including smart mobility [57], opinion dynamics [58], distributed control [59], epidemiology [60], building evacuation [61, 62], human-unmanned vehicle interaction [63], and multiple robot coordination [64–66].

Game theory offers substantial benefits compared to alternative modeling methods, such as reactive strategies [20, 67, 68] and learning schemes [22–25]. With respect to the former, game theory is able to perform motion prediction and anticipation of the behavior of other humans, typical of human decision-making in social contexts [69]. Compared to the latter, it overcomes their distinctive lack of explainability, generalization, and the need for large training data sets. For a thorough discussion of socially aware navigation literature, please refer to the subsequent section.

Nevertheless, despite the well-known ability of the game theory to model different aspects of human behavior, only a few efforts have attempted to incorporate it into a mobile robot motion. This limited literature could be attributed to the inherent non-

negligible computational complexity of the game theory which therefore leads to the choice of computationally lighter solutions but at the expense of human prediction accuracy.

The main bottleneck of the algorithms that adopt the game-theory is related to the estimation of the Nash equilibrium- the condition in which no agent has an incentive to unilaterally change its own action (or strategy) if the other players do not change theirs. Despite this, in the design of the two algorithms presented in this thesis, we adopt the game logic and we postulate that the players, involved in the game, tend to reach a Nash equilibrium. This hypothesis is justified by the study conducted by Turnwald [69], where the main result reveals that the Nash equilibrium solution leads to trajectories that closely resemble human decision-making patterns. Nevertheless, to handle the Nash equilibrium calculation in real-time in the two algorithms, we used solutions that balance computational efficiency with the capacity to model human behavior. In particular, in the first algorithm, each player (human or robot) interacts only with those players whose initially estimated trajectories intersected, as detailed in Chapter 3. This approach limits the number of players each individual considers, reducing computational demands. Additionally, the actions available for each player are discretized and restricted to seven, further simplifying the decision-making process.

The second algorithm employs a similar strategy, limiting the number of choices for each player to four to enable real-time calculation of Nash equilibrium.

Thus, the approaches presented above allow a trade-off between the computational feasibility and the ability to incorporate human anticipation into the navigation algorithm.

Moreover, continuous advances in the computational power of microcontrollers have opened up new possibilities to implement even computationally demanding algorithms on modestly sized mobile robots. This technological advance paves the way for exploring the potential of game theory based approaches in this domain.

For these reasons, this thesis presents two novel socially-aware navigation algorithms exploiting the game-theory methodology. In particular, the first algorithm leverages pure game theory (Chapter 3), while the second adopts a hybrid approach, combining the game theory principles with the well-known social force model algorithm (Chapters 4, 5, and 6).

2.2 Literature review

Different surveys have proposed a wide range of solutions to classify the socially-aware navigation approaches [26, 70]. In particular, in [70] authors enhance the research done in [26] with the recent growth of deep learning approach. Following the classification suggested in [70], the state of the art can be classified into: reactive based, predictive based, model based and learning based. However, this type of classification allows some algorithms to fit into multiple categories at the same time. Therefore, in this thesis we chose to classify algorithms (according to [71]) in a more general way: *model-based* and *learning-based*. *Model-based* algorithms rely on human-motion mathematical models, like geometric rules or physics equations, to enable the navigation robot process. On the other hand, *learning* algorithms, learn from real human motion data, allowing them to better mimic human behavior in navigation.

At the state of the art, many authors have treated the problem using the *model-based* approach. A prominent representative in this category is the Social Force Model (SFM). It leverages attractive and repulsive forces between agents (detailed in Chapter 4) to simulate human-like movement based on the laws of Newtonian mechanics. Motivated by the model's potential for broader applicability, subsequent research has focused on expanding its capabilities through the addition of new forces. An example is in [72], where authors took a step forward by extending the social force model to design a robot's motion plan that seamlessly accompanies a person walking alongside. This innovation involved introducing a force that specifically accounts for the dynamics of human-robot interaction. To improve robot navigation in social settings, the authors in [73] propose a novel approach that integrates human intention into the social force model. By leveraging body pose and face human orientation, the model infers human intent, enabling robots to react more appropriately in dynamic social situations.

The Velocity Obstacle (VO) planner, introduced by Fiorini and Shiller in 1998 [48], generates avoidance maneuvers by determining a permissible velocity for the robot outside the collision cone. The latter represents the space of velocities that could lead to a collision with the moving obstacle. However, VO does not consider the reciprocal mutual avoidance typical of human behaviour during navigation. Thus, recognizing this limitation, Berg et al. [49] introduced Reciprocal Velocity Obstacle (RVO). While this approach effectively prevents collisions under specific circum-

stances, it fails to offer a definitive guarantee for all scenarios (such as when both agents choose the speed inside each other's RVO), lacking the generality required for comprehensive collision avoidance. To overcome this problem, RVO was further refined in [21] with Optimal Reciprocal Collision Avoidance (ORCA).

The algorithms presented above adopt a reactive navigation strategy, primarily reacting to the immediate presence of pedestrians through obstacle avoidance maneuvers rather than anticipating their future movements. However, within the *model-based* category, there also exist approaches that overcome the limitations of reactive logic by incorporating predictions of human future movements within a specific time horizon. An example of such approach is the adoption of game theory to design motion planning for a mobile robot. Despite the well-known ability of the game theory to model different aspects of human behavior, only a few efforts have attempted to incorporate it into a mobile robot motion. In this context, Turnwald et al. [69] modelled human navigation as a non-cooperative game. They evaluated five different cost functions and conducted real-world experiments with two participants to identify the most effective one. Notably, their findings suggest that path length minimization produced the best results. However, the core contribution of their work lies in real-world experiments on how pedestrians choose trajectories based on achieving a Nash equilibrium. The same author in [74], investigates how the game-theoretic human model can be further adapted for designing a human-like motion planning.

While the *model-based* approaches have impressive advantages like rapid implementation without a training process with data and the generalizability to a more complex environment, they face certain limitations. Firstly, the lack of human-like behavior training with real data might lead to unnatural navigation patterns, potentially causing confusion or discomfort in shared spaces. Secondly, fine-tuning the model's parameters for optimal performance in specific environments can be a complex and time-consuming task.

On the other hand, the continuous advancements in deep learning technology and the integration of deep neural networks are contributing to the growing popularity of *learning-based* approach. The latter can be divided into three classes according to their different functionalities: supervised learning, deep reinforcement learning and inverse reinforcement learning.

The supervised learning leverages real-world pedestrian trajectory data to train algorithms that capture and reproduce the social interactions observed in human navigation. Based on this line of research, Singh et al. [75] and Xie et al. [76] demonstrate the effectiveness of supervised learning for robotic navigation in dynamic environments.

The deep reinforcement learning algorithms mimic the way animals and humans learn, adapting their behavior based on rewards. These latter are mathematically represented as a score (reward function) that provides feedback on how well an agent's interaction with the environment aligns with its objectives. An example of this method is adopted in [77, 55, 78]. However, designing reward functions for deep reinforcement learning in robotics could be difficult, especially in complex scenarios [70]. Thus, inverse reinforcement learning offers a solution by learning from expert demonstrations to automatically deduce the underlying reward structure, which can then be used by deep reinforcement learning for training social-aware navigation policies [79]. Recent works by [80, 81] prove the successful application of this method in the robot navigation context.

Unlike *model-based* approaches, *learning-based* methods leverage real-world trajectory data to achieve behavior closer to human navigation. However, this advantage comes at the cost of extensive data requirements for training and limited generalizability beyond the specific training scenarios. Moreover, the limited interpretability of the output of these methods presents a significant challenge, creating difficulties in comprehending the underlying causes of specific behaviors. Consequently, debugging and diagnosing issues become more complex posing a major drawback compared to *model-based* methods. Limited interpretability of the outputs in some cases may lead to unexpected and potentially dangerous behaviors in human-shared environments. Careful design of recovery and emergency mechanisms is crucial to ensure the robot's movements remain safe and socially acceptable under all circumstances.

2.3 Performance metrics

To enhance the robot's acceptance, navigation algorithms should ensure both their effectiveness in achieving goals and their ability to provide a natural and comfortable experience for humans (for details see Section 1.1). This necessitates the use of

Symbol	Meaning
$\mathbf{p}_i$	position of the agent i
$\mathbf{p}_j$	position of the agent j
$\mathbf{v}_i$	speed of the agent i
T_{goal}	number of time steps needed to reach the goal
$\theta_i(t)$	orientation of the agent i at time t

Table 2.1 Notation of the performance metrics.

precise performance metrics that *quantitatively* assess these aspects. Several state-of-the-art performance metrics, such as those proposed by Biswas et al. [82] and Gao et al. [27], have been developed for this purpose.

This section focuses on the chosen performance metrics adopted in this thesis. Since some of these metrics will be also applied to human trajectory evaluation (Chapter 3), henceforth the term *agent* is used to refer to both robot and human.

The notation used in this section is summarized in Table 2.1.

The performance metrics used to assess the trajectories' performance are:

- ***Path Length Ratio (PLR)***

  This performance metric measures the ratio between the direct line-of-sight distance between the starting and ending points of a path, and the actual distance traversed by the agent between those two points:

$$PLR = \frac{\|\mathbf{p}_i(T_{goal}) - \mathbf{p}_i(0)|}{\sum_{t=1}^{T_{goal}} \|\mathbf{p}_i(t) - \mathbf{p}_i(t-1)\|} \tag{2.1}$$

  where T_{goal} is the number of time steps needed for the agent to reach its goal. Hence, this performance metric takes values in the range [0,1]. Generally, a high PLR is desirable, signifying that the agent tends to reach its destination while minimizing the length of the path. By measuring this metric, we can analyze how efficiently and successfully the agent navigates towards achieving its goal;

- ***Average Speed (AS)***

  This metric represents the average speed of the agent along the entire trajectory, a factor used sometimes in evaluating its overall performance:

$$AS = \frac{\sum_{t=1}^{T_{goal}} \mathbf{v}_i(t)}{T_{goal}} \tag{2.2}$$

However, this metric may also capture perceptions of *comfort*. Studies suggest a negative correlation between robot speed and perceived safety, potentially impacting comfort evaluations [83];

- ***Closest Pedestrian Distance (CPD)***

  This metric denotes the distance between the agent i and the nearest agent j during the entire considered trajectory:

$$CPD = \min_{t,\ j} \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| \tag{2.3}$$

  An increased *CPD* value signifies that the agent i has a propensity to stay further away from the other agents. This contributes to increasing the level of *comfort* for humans;

- ***Path Regularity (PR)***

  This metric evaluates *naturalness* by measuring the normalized rotations made by the agent during navigation.

  Such metric is computed as follows:

$$PR = 1 - \frac{\sum_{t=1}^{T_{goal}} |\theta_i(t) - \theta_i(t-1)|}{PI_{max}} \tag{2.4}$$

  where $\theta_i(t)$ denotes the orientation of the agent at a given time instant t. The denominator PI_{max} normalizes the path irregularity factor and is calculated as follows:

$$PI_{max} = \max_{\substack{algorithm_1,\\ algorithm_2,\\ algorithm_3}} \sum_{t=1}^{T_{goal}} |\theta_i(t) - \theta_i(t-1)| \tag{2.5}$$

  where the maximization should be intended across all considered algorithms.

  The *PR* assumes values within the range of [0,1]. A value of 1 signifies a direct path from the agent's starting point to the goal. On the other hand, a value

approaching 0 indicates a significant number of rotations during navigation. For our purpose, it is preferred to obtain a *PR* value closer to 1, to have a smooth motion with few sudden changes in direction.

Each algorithm, developed in this thesis, is evaluated using a tailored set of presented metrics, carefully chosen to align with the specific context, algorithm objectives, and intended evaluation scope.

Chapter 3

Game theoretical (GT) trajectory planning to enhance robot social acceptance

Since humans and robots are increasingly sharing portions of their operational spaces, experimental evidence is needed to ascertain the safety and social acceptability of robots in human-populated environments. Although several studies have aimed at devising strategies for robot trajectory planning to perform safe motion in populated environments, a few efforts have measured to what extent a robot trajectory is accepted by humans. In this chapter, we present a navigation system for autonomous robots that ensures safety and social acceptability of robotic trajectories. We overcome the typical reactive nature of state-of-the-art trajectory planners by leveraging non-cooperative game theory to design a planner that encapsulates human-like features of preservation of a personal space, recognition of groups, sequential and strategized decision-making, and smooth obstacle avoidance. Social acceptability is measured through a variation of the Turing test administered in the form of a survey questionnaire to a pool of *691* participants. Comparison terms for our tests are a state-of-the-art navigation algorithm (Enhanced Vector Field Histogram, VFH) and purely human trajectories. The experiment revealed that participants easily recognized the non-human nature of VFH-generated trajectories, but they had difficulty distinguishing between game-theoretical trajectories and human-generated trajectories.

The chapter is structured to provide a comprehensive overview of the study. In particular, firstly we outline the methodological approach employed in this study and we situate this approach within the existing body of research to understand its relative novelty and contribution to the field. Then, we present the methodology employed to design the algorithm, providing a detailed description of the technical aspects. Following this, we present the results of the Turing test, offering insights into the algorithm's performance. Finally, we conclude the chapter with a discussion that encourages further exploration.

For a comprehensive review of the key concepts and current state-of-the-art in socially-aware navigation, please refer to Chapter 2.

3.1 Overview of the approach and main contributions

In this chapter, we present a socially-aware robot navigation strategy that accurately models human behavior using game theory (see Figure 3.1 for a graphical abstract of the procedure). In particular, our approach uses non-cooperative game theory [56] to model the navigation behavior of multiple humans in populated environments, positing that conditions of safe navigation, adherence to social norms, and psychological comfort correspond to a Nash equilibrium in the proposed game-theoretical model. Differently from [69, 74], our model contemplates more than two players –a feature that is essential to model populated environments. The human motion model informs the design of a robotic trajectory planner, whereby the robot tends to mimic human behavior during motion and interaction in a populated environment.

Our work marks an important milestone in the field of social robotics. It provides an efficient, social-aware motion planning framework that encapsulates realistic features of human crowds, remarkably enhancing the social acceptance of the planned trajectories. Namely, we incorporate the human personal space (i.e. the region around the human in which others cannot intrude without causing discomfort) [29], the recognition of human groups [84], the sequential decision-making typical of human beings [85], and a natural human-obstacle interaction [86] –features that are often missing in many approaches, including those based on game theory [74].

The methodology proposed in this chapter is generally applicable to any kind of mobile robot. To avoid confounds related to the choice of specific hardware setup

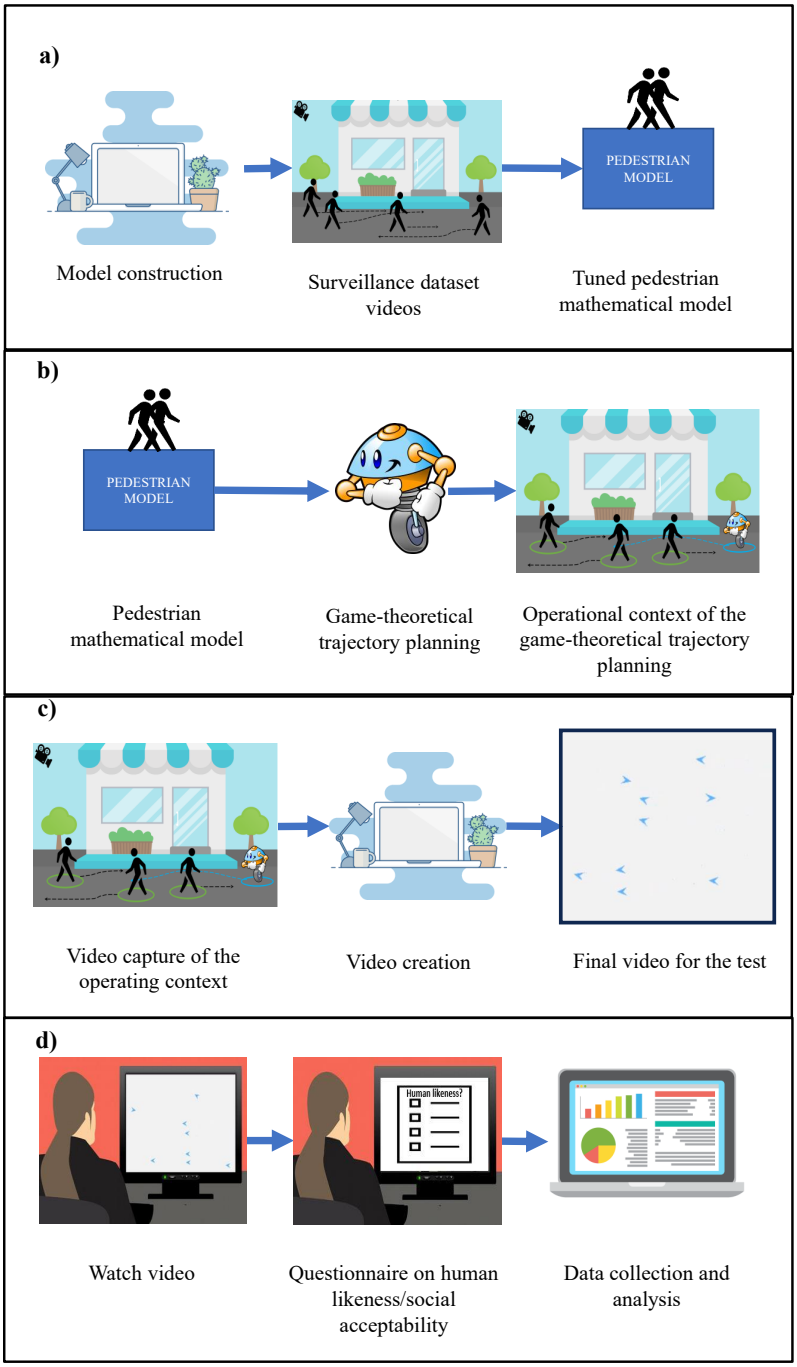


Fig. 3.1 Graphical abstract of the procedure. **a)** Construction of the game-theoretical model for human motion; **b)** creation of the game-theoretical trajectory planner based on the model previous designed, creation of the virtual environment and evaluation of the performance metrics; **c)** creation of the videos with pedestrians and the robot controlled by our game-theoretical trajectory planner; **d)** survey questionnaire and data collection.

and focus on the assessment of human perception of the robot's motion, validation is executed on virtualized environments, where the humanly populated scene is extrapolated from surveillance videos. Three different experimental conditions are considered: the first involves only human subjects, the second contains a virtualized mobile robot programmed through the state-of-the-art Enhanced Vector Field Histogram (VFH [87]) algorithm moving through the population, the third replaces the VFH algorithm with our game-theoretical approach.

Across the three experimental conditions, we perform a twofold validation of our approach: first, we evaluate performance metrics typical of path planning (path length ratio, path regularity, and distance to the closest pedestrian), and then we analyze the results of a survey questionnaire to directly assess social acceptability by human subjects. To this aim, we administered a variant of the Turing test to a pool of 691 volunteers, who evaluated the human likeness of three sets of videos corresponding to the three scenarios explained above. To conceal the appearance of the agents, we masked humans and robots by replacing them with arrows so that the volunteers could not distinguish between them.

Evidence from our experimental campaign reveals that trajectories generated by our game-theoretical approach exhibit performance metrics that are efficient and closer to those achieved by human subjects than VFH. Moreover, the outcome of the survey questionnaire highlights the superior acceptability of game-theoretical-generated trajectories with respect to those generated through VFH.

3.2 Methodology

Figure 3.1 schematizes the proposed procedure for the realization and validation of our game-theoretical framework for the social acceptability of robotic trajectories. The methodology can be subdivided into four main logical phases, corresponding to the panels in the figure. First, a game-theoretical model of pedestrian motion is devised and its parameters are tuned based on the analysis of human motion videos (panel (a)). Second, a robotic trajectory planner informed by the game-theoretical pedestrian model is realized. The robot is deployed and operated in a virtual humanly populated environment, where humans execute real trajectories extracted from videos. In this phase, three important performance metrics in robotic trajectory planning (path length ratio, path regularity, and distance to the closest pedestrian) are evaluated

across the three experimental conditions (panel (b)). Third, the virtual environments containing humans and the robot are processed and prepared to be administered for the validation questionnaire (panel (c)). Finally, the questionnaire is administered and the results are collected and analyzed (panel (d)). In the following, the main components that constitute our methodology will be illustrated in detail.

3.2.1 Game-theoretical model

Assumptions

Let us start with a description of all the assumptions supporting our game-theoretical model for human motion. To improve readability, here and henceforth we will refer to human subjects as *agents*. This term will be also used for the robot when no distinction between the two categories is required.

All pedestrians are *rational* agents with *common knowledge* moving in a 2D *populated dynamic* environment.

Rational behavior entails that agents only aim to reach their own *individual* motion goal (i.e. the location to which the agents wish to go). In mathematical terms, this translates into a minimization of an individual cost (equivalently, a maximization of an individual benefit), such as their overall path length [88] or energy consumption [89]. Practically, agents continuously update their navigation behavior while walking in populated environments, based on the observation and possible prediction of the motion of the surrounding agents.

The possession of *common knowledge* by agents in our game-theoretical model implies that all agents have the same knowledge about what actions can be performed to reach their final goal and how other pedestrians behave while walking.

Such an assumption is reasonable when dealing with models of human traits, as individuals commonly learn these skills by experience during everyday life [74].

Specifically, we consider a *populated dynamic environment*, possibly busy, but not crowded, such as typical streets occupied by pedestrians walking on sideways, or populated indoor spaces, such as hotel halls [74]. We suppose that the environment contains static obstacles, that have to be avoided by agents in a *natural* manner. Our approach is based on a microscopic modeling strategy, whereby a single individual

is mapped onto a single software agent, which mimics the individuals' decision and their interactions.

Game description

The proposed model for pedestrian motion is a non-cooperative, static, perfect information, finite, and general-sum game with many players (or agents).

In our model, each agent aims at reaching its own goal *individually*, but the minimization of its individual cost does not exclude the possibility of collaborating with other agents, should this help to attain *individual* goals [90] as well.

Our model recognizes as *groups* those pedestrians that move close to each other keeping a similar direction of motion. These groups of agents are considered as single players, whereby members of the group share a common strategy and a common motion pattern. This last assumption practically entails that the navigation strategy of the robot in avoiding human groups would treat them as a compact group of people that cannot be split to better attain its own navigation goal.

The game is *static* in the sense that agents move and take decisions *simultaneously*, it is based on *perfect information*, that is, each agent knows the current and the previous actions of all agents, e.g. via direct observations.

The game is also *finite*, i.e., the game has a *finite* number N of agents belonging to the agent set $\mathcal{N}$, where each agent $i \in \mathcal{N}$ can choose among a *finite* number of actions available, defined with the action set Θ, which is supposed to be common to all agents. In particular, we indicate with $\theta_i(t) \in \Theta$ the action executed by agent i at the discrete time t. In our application, the execution of action $\theta_i(t)$ corresponds to a motion of agent i in the 2D plane at constant velocity v and constant heading $\theta_i(t)$ over the whole discrete time step Δt. We assume that agents have a bounded visibility angle and the possible action $\theta_i(t)$ is designed to uniformly partition such an angle. We denote with $p_i \in \mathbb{R}^2$ the position of agent i in the 2D environment, with respect to a fixed orthogonal reference frame.

Moreover, the proposed model is a *general-sum* game, i.e., the sum of all gains and losses of the utility functions over all agents is *not necessarily* equal to zero.

Similar to [69], we postulate that, in such a navigation task, agents tend to reach a Nash equilibrium – the condition in which no agent has an incentive to unilaterally

change its own action (or strategy) if the other agents do not change theirs. In other words, a Nash equilibrium occurs when each agent achieves its best response, i.e., its minimum individual cost, given the actions of the other agents. In general, however, existence and uniqueness of a Nash equilibrium is not guaranteed in our setup, and its analytical characterization is almost always impossible to have, thus making numerical approaches for an approximate computational necessary. Here, the Nash equilibrium is approximately computed via the *sequential best response* approach [91].

Let us explain the idea of the sequential best response for two agents, A and B: agent A observes the motion of agent B and then solves an optimization problem to determine its own strategy, given the latest observed strategy of agent B. Afterwards, a check action is performed, verifying if the strategies of both agents are the same as those computed in the previous iteration; in such a case, the game has reached a Nash equilibrium. Otherwise, agent B computes its optimal strategy, given the latest observed strategy of agent A. The procedure is applied iteratively, until the equilibrium condition is met. The same strategy identically extends to N agents.

Our modeling procedure assumes that all the agents in the planar space play the game mentioned above. After the model has been identified, we will use it to control a single, synthetic agent to navigate through the populated environment. Such an agent is called *robot player*.

Optimization problem

The sequential best response approach in our game-theoretical model for the human motion in a populated environment requires the solution of a set of interdependent optimization problems, one for each agent moving in the environment. The goal of the optimization problem for each agent i is to find the best sequence of actions, $\boldsymbol{\theta}_i^* = (\theta_i(t), \theta_i(t+\Delta t), \theta_i(t+2\Delta t), \ldots, \theta_i(t+T\Delta t))$, over a finite prediction horizon $T\Delta t$, given the actions of the other agents. Without loss of generalization and to improve readability, here and henceforth we assume a unitary discrete-time step, i.e., $\Delta t = 1$.

All agents seek for the Nash equilibrium by applying the sequential best response strategy, solving their own optimization problem based on the observed behavior of

the rest of the population. We define the optimization problem for each agent $i \in \mathcal{N}$ as

$$\boldsymbol{\theta}_i^* = \min_{\boldsymbol{\theta}_i} \quad J(\boldsymbol{\theta}_i) \tag{3.1a}$$

$$\text{s.t.} \quad \left\| p_i(t, \theta_i(t)) - p_j(t) \right\|_2 \geq \beta \quad \forall\, t, \forall i, j \in \mathcal{N}, i \neq j \tag{3.1b}$$

$$p_i(t, \theta_i(t)) \notin \mathcal{O}_{\text{obs}} \quad \forall\, t, \forall i \in \mathcal{N} \tag{3.1c}$$

with

$$p_i(t, \theta_i(t)) = p_i(t-1, \theta_i(t-1)) + \Delta p(\theta_i(t), v). \tag{3.2}$$

The cost function $J(\boldsymbol{\theta}_i)$ in (3.1a) is defined as

$$J(\boldsymbol{\theta}_i) = \Phi_{\text{goal}}(\boldsymbol{\theta}_i) + \Phi_{\text{smooth}}(\boldsymbol{\theta}_i) + \Phi_{\text{obs}}(\boldsymbol{\theta}_i), \tag{3.3}$$

where the three summands are defined as follows:

(i) The term $\Phi_{\text{goal}}(\boldsymbol{\theta}_i)$ tends to reduce the overall path length for each agent i and, hence, models the goal-oriented attitude of the agent:

$$\Phi_{\text{goal}}(\boldsymbol{\theta}_i) = \sum_{t=1}^{T} \gamma(t) \| p_i(t, \theta_i(t)) - p_i^* \| \tag{3.4}$$

with $\gamma(t)$ being a time-variant weight factor; $p_i(t, \theta_i(t))$ is the estimated position of agent i at time t, considering a constant speed modulus v and the heading control action $\theta_i(t)$ applied at time t, computed using the kinematic update Equation (3.2); and p_i^* is the estimate of agent i's goal in the time horizon T. In the absence of an explicit definition of a pedestrian's goal, we assume that, within the horizon $[t, t+T]$, the goal of agent i lays on a straight line starting in $p_i(t)$ and oriented along the observed agent heading at time t. Under these assumptions, the practical meaning of the time horizon T is the estimate of the time interval within which a pedestrian sets up and maintains their walking goal.

(ii) The term $\Phi_{\text{smooth}}(\boldsymbol{\theta}_i)$ penalizes excessive rotations, thus promoting smooth trajectories. In fact, during navigation, humans tend to avoid too many changes of

orientation to minimize their energy consumption [89]:

$$\Phi_{\text{smooth}}(\boldsymbol{\theta}_i) = \sum_{t=1}^{T} (1-\gamma(t))|\theta_i(t) - \theta_i(t-1)| \tag{3.5}$$

where $\theta_i(t)$, $\theta_i(t-1)$ are the orientation of the agent at time t and $(t-1)$, respectively. We observe that the term $\Phi_{\text{smooth}}(\boldsymbol{\theta}_i)$ is weighted in a complementary fashion to $\Phi_{\text{goal}}(\boldsymbol{\theta}_i)$, to satisfy the assumption (further detailed in the Implementation section) of their relative importance as long as the agent approaches its target.

(iii) The term $\Phi_{\text{obs}}(\boldsymbol{\theta}_i)$ tends to optimize the natural interaction with static objects. In fact, humans tend not to walk too close to static obstacles, unless it is necessary. For this reason, we model this behavior as a *soft* constraint:

$$\Phi_{\text{obs}}(\boldsymbol{\theta}_i) = \sum_{t=1}^{T} \frac{\rho}{\|p_i(t,\theta_i(t)) - p_{\text{obs}}\|} \tag{3.6}$$

where ρ is a weighting factor and the denominator in (3.6) is the distance between the agent position $p_i(t,\theta_i(t))$ and the closest *static* obstacle p_{obs} at time t. The exact procedure to compute p_{obs} will be explained later. Practically, Equation (3.6) penalizes small distances between an agent and *static* obstacles.

The inequality in (3.1b) is a hard constraint imposing to avoid other agents, assuming a circular region around agents as their personal space [29] to be avoided. In this way, agent i is required to maintain at least a minimum distance β with other agents in the observer scenario. Constraint (3.1c) models the avoidance of static obstacles by imposing that the position $p_i(t,\theta_i(t))$ is outside the obstacle space $\mathscr{O}_{\text{obs}}$, defined as a subset, possibly disconnected, of the 2D planar space, occupied by obstacles, where motion of agents is forbidden.

Equation (3.2) formalizes the kinematic update of the position of agent i at time t, subject to a heading command $\theta_i(t)$, at a constant velocity v.

3.2.2 Validation of the game-theoretical model

The proposed game-theoretical human motion model is validated by conducting a qualitative comparison between generated trajectories and human ones, observed in open-source surveillance videos [92, 1]. These surveillance videos, used to validate

the proposed model, show a typical urban scenario in which multiple agents walk interacting with each other and avoiding static obstacles. Figure 3.2 illustrates the frames, randomly selected, of the surveillance videos of two different scenarios. Specifically, Figure 3.2 compares real trajectories executed by humans (Figs. 3.2**a** and 3.2**c** with the estimated trajectories generated for all agents by the proposed model solving our game-theoretical problem (Figs. 3.2**b** and 3.2**d**).

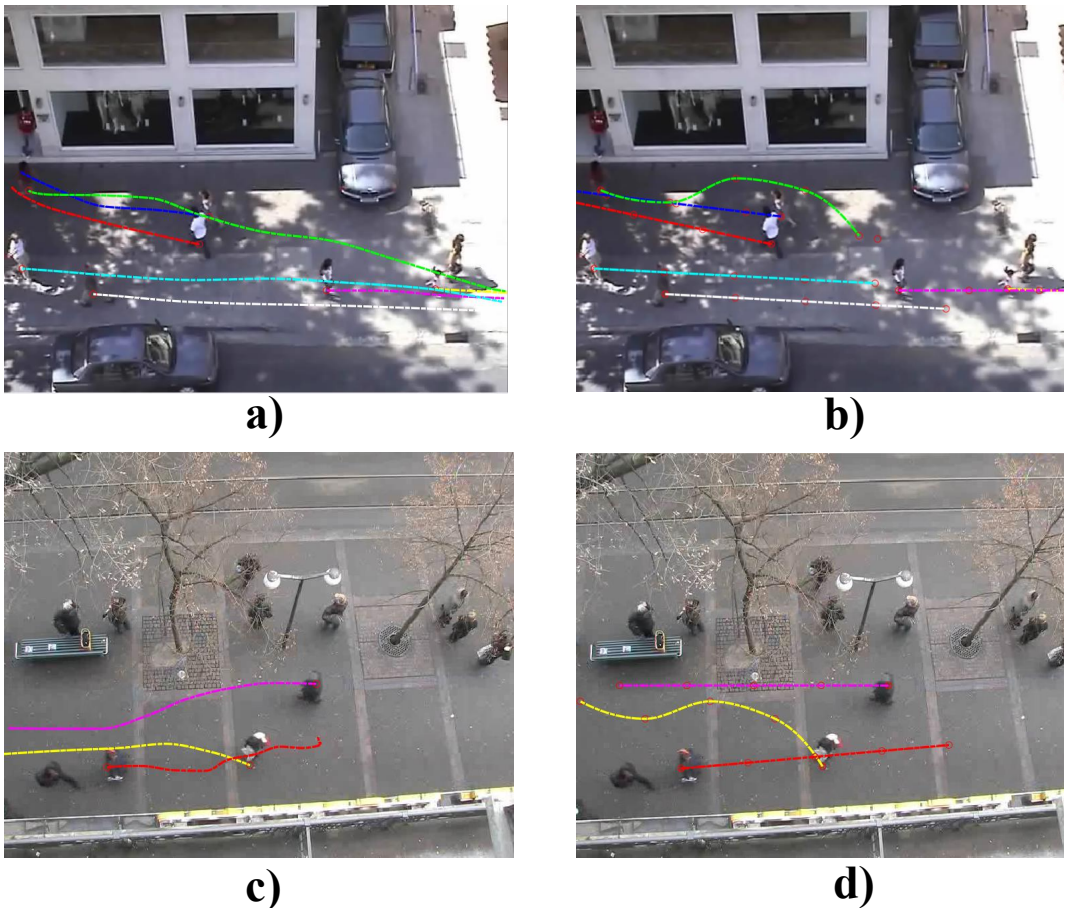


Fig. 3.2 Validation of our human motion model based on game-theoretical approach with open-source surveillance videos [1].
a)-c) Real human trajectories; **b)-d)** Trajectories output of the game-theoretical model.

We observe that, in both the illustrated scenarios, our game-theoretical approach generates collision-free trajectories (Figs. 3.2**b** and 3.2**d**) that are smooth and resemble those executed by their human counterparts. However, we note that the trajectories generated by our algorithm exhibit a sharper reaction than humans in the vicinity of surrounding agents. This is evident while comparing Figure 3.2**a** and

Figure 3.2**b**, focusing on the interaction between the green trajectory and the blue one. A comparable circumstance can be observed in Figure 3.2**c** and Figure 3.2**d**, with reference to the yellow trajectory. This phenomenon is most likely caused by the discrete action set associated with each agent. Notably, in our implementation an agent can choose one out of seven possible headings inside their own visibility zone, resulting in a resolution of $\pm\pi/6$ rad, in the attempt of minimizing the corresponding cost function. On the other hand, human subjects can select their heading over an infinite set.

A further cause of discrepancy between human and game-theoretical trajectories resides in the kinematic update of the agent position in Equation (3.2) –a linear update with constant heading and velocity over the whole sampling step– and the estimation of the human target, assumed to be constant over an interval of duration T –actually an unknown, subject to the very stochastic nature of human behavior.

3.2.3 Algorithm

The game-theoretical model of pedestrian motion described above is used to inform a robotic trajectory planner for autonomous robots moving in populated environments.

Algorithm 1: Main algorithm

Initialization:
$p_{\text{robot}} \leftarrow$ InitializeRobotPosition
repeat
  GroupRecognition
  $\boldsymbol{\theta} \leftarrow$ FirstEstimation
  foreach *agent i (robot included)* **do**
    $C_{\text{obs}} \leftarrow false$
      [flag defining the collision with obstacles for agent i]
    $C_{\text{agents}} \leftarrow false$
      [flag defining the collision between agents for agent i]
    $C_{\text{obs}}, C_{\text{agents}} \leftarrow \text{CheckCollision}(i, \boldsymbol{\theta})$
    $\boldsymbol{\theta_i} \leftarrow \text{ComputeSolution}(i, \boldsymbol{\theta}, C_{\text{obs}}, C_{\text{agents}})$
  $p_{\text{robot}} \leftarrow$ UpdateRobotPosition
until $p_{\text{robot}} = p_{\text{goal}}$;

The main steps executed by the proposed trajectory planner are described in Algorithm 1. First, the robot position (p_{robot}) is initialized using the function

Algorithm 2: The ComputeSolution function

```
ComputeSolution(i, θ, C_obs, C_agents)
    if C_agents then
        θ_i^gt ← GTPlanner(θ) [Solution to Algorithm 3]
        θ_i^dec ← Decelerate
        if Cost(θ_i^gt) ≤ Cost(θ_i^dec) then
            θ_i ← θ_i^gt
        else
            θ_i ← θ_i^dec
    else
        if C_obs then
            θ_i ← IndividualOptimization
    return θ_i
```

InitializeRobotPosition. Then, the algorithm executes an iterative procedure that stops when the robot reaches its target position (p_{goal}). Here, we will refer to both humans and the robot with the term "agent". Each iteration performs five main steps: recognition of groups of humans (GroupRecognition), first estimation of trajectories for all agents (FirstEstimation), collision checking between agents and with obstacles (CheckCollision), computation of the agent trajectory (ComputeSolution), and update of the robot position using the computed trajectory (UpdateRobotPosition). This iterative procedure predicts the agents' motion and generates the robot's optimal trajectory over the fixed time horizon T, by applying the strategy detailed below. After such an optimal trajectory for the robot is computed, only the action corresponding to the first time step is actually applied to the robot and the process is repeated until the robot reaches its goal.

In the following, each step of the Algorithm 1 is detailed:

- GroupRecognition. The algorithm performs the *group recognition* of agents considering the observed orientation of each agent, and the distances between them. In fact, a group is typically moving maintaining a common orientation and keeping a distance between agents shorter than the personal space typical of the single agent. Upon recognition, groups are considered as *unique entities* and treated as single agents in the subsequent phases.

- FirstEstimation. A preliminary estimation of all agents' trajectories (i.e., $\boldsymbol{\theta}$) is performed, projecting hypothetical rectilinear trajectories over the interval T.
- CheckCollision. Given the trajectories of all agents ($\boldsymbol{\theta}$), previously estimated by the FirstEstimation, the CheckCollision function detects the possible occurrence of collisions between an agent i with obstacles and other agents, activating the flag variables C_{obs} and C_{agents}, respectively. Moreover, we refer to the occurrence of a collision with other agents also when the individual personal space of an agent is violated.
- ComputeSolution. Considering the estimated trajectories ($\boldsymbol{\theta}$), and the flags C_{obs} and C_{obs}, Algorithm 2 computes a solution to the motion planning problem for an agent i selecting one of the possible cases:

  (i). if a collision with other agents is envisaged, two alternative solutions are evaluated. Hence, the solution that involves the lowest cost of Equation (3.3) will be selected.

  The first solution ($\boldsymbol{\theta}_{\boldsymbol{i}}^{\text{gt}}$) is computed using the strategy defined in Algorithm 3, where trajectories are generated seeking for a Nash equilibrium solution of the game presented in the *Game description* section.

  The second solution is computed through the Decelerate function, which evaluates the opportunity to *decelerate* –a typical human behavioral trait in navigation– to avoid the collision with other agents. In particular, after identifying the discrete time step t at which a collision between agent i and other agents is envisaged to occur, the cost associated with sixteen different deceleration patterns is evaluated using the cost function (3.3), provided that constraints in Equations (3.1b) and (3.1c) are satisfied;

  (ii). if an agent is envisaged to collide with a *static* obstacle (C_{obs}), the agent solves its individual optimization problem described above (without playing the game and, hence, not seeking for the Nash Equilibrium);

  (iii). if no collision between agents or static obstacles is envisaged, trajectories are kept linear, maintaining the current heading and constant velocity, practically implementing what was already computed in the FirstEstimation procedure.

- UpdateRobotPosition. Considering the computed trajectory of the robot, the action corresponding to the first time step is executed and the robot position is updated using Equation (3.2).

Algorithm 3: Nash trajectory computation

Initialization:
$k \leftarrow 1$
[iteration index]
$\boldsymbol{\theta}^k \leftarrow \mathbf{0}$
[straight paths for all agents as FirstEstimation]
$i \leftarrow 1$
[agent index]
Iterate until convergence:
$\bar{p}_j^k \leftarrow$ (3.2), given $\boldsymbol{\theta}^k$, for all j
[present and future predicted positions of all agents]
$\boldsymbol{\theta}_{\boldsymbol{i}}^{k+1} \leftarrow$ solution to Eq. (3.1a–3.1c), given $\left(\bar{p}_j^k\right)_{j \neq i}$
[best response to all other agents]
if $i < \mathcal{N}$ **then**
  $i \leftarrow i+1$
  [move on to next agent]
else
  $i \leftarrow 1, k \leftarrow k+1$
  [move on to next iteration]
end

Implementation The algorithm presented above has been implemented in Matlab and the main implementation choices are discussed in what follows.

The discrete time step has been set to $\Delta t = 1.2$s. The time horizon for optimization has been set to $T = 4$, that is, 4.8 seconds. In the following, we opted to keep a unitary discrete time step, to enhance readability.

As previously stated, each agent can execute actions taken from an action set Θ of finite size. Specifically, in our implementation, each agent has seven possible actions for $\theta_i(t)$, which represents the heading within the agent *visibility* zone. Namely, $\theta_i(t)$ is updated as $\theta_i(t) = \theta_i(t-1) + u(t-1)$, where $u(t-1)$ takes values in the finite set $\Theta = \{-\pi/2, -\pi/3, -\pi/6, 0, \pi/6, \pi/3, \pi/2\}$ rad. We remark that we limited the cardinality of Θ to seven, pursuing a trade-off between satisfactory performance and reasonable computational complexity of the algorithm.

In Equation (3.1b), the β parameter is set considering the Hall convention [29] that posits the existence of a personal space of circular shape that ensures comfort conditions for human navigation. The value of β has been estimated through the analysis of the open-source surveillance videos [92, 1].

In Equation (3.3), the term ($\Phi_{\text{obs}}(\boldsymbol{\theta}_i)$) can be neglected if the first estimation of the agent trajectory does not intersect any static obstacle. Otherwise, $\Phi_{\text{obs}}(\boldsymbol{\theta}_i)$ in Equation (3.6) is computed referring to the closest obstacle, toward which the agent is projected to collide. Then, the closest point of such obstacle to the agent position is computed (p_{obs}). To reduce the computational load, obstacles are mapped into a discrete spatial map overlapping with the 2D environment. The map consists of a rectangular matrix of 576x720 cells, which are marked as being occupied by an obstacle or free from them. Each cell covers approximately a square of 1.8x1.8 cm.

The weight $\gamma(t)$ in Equations (3.4) and (3.5) is selected as a time-varying term that is used to balance the relative importance of terms $\Phi_{\text{goal}}(\boldsymbol{\theta}_i)$ and $\Phi_{\text{smooth}}(\boldsymbol{\theta}_i)$ over the optimization horizon T. This choice emerges from the analysis of the available surveillance videos, where we observed that the minimization of the distance to the goal typically prevails on the smoothness requirement as long as the agent gets closer to their goal, and vice versa. Considering $T = 4$ time steps, we chose the following sequence for $\gamma(t)$, starting from a generic time instant t^*: $\gamma(t^*) = 0.6$, $\gamma(t^*+1) = 0.7$, $\gamma(t^*+2) = 0.8$, $\gamma(t^*+3) = 1.0$.

3.2.4 Trajectories generation for performance metrics

We designed trajectories for a preliminary *quantitative* assessment using performance metrics in three experimental conditions, which differ for the algorithm governing the motion of a selected agent (i.e., either a robot or a human being): in the condition *humans only* (HO), all the agents were human beings moving in a real environment; in the condition *humans and GT* (GT), one of the agents was controlled by our game-theoretical algorithm, while the other agents were human beings; and in the condition *humans and VFH* (VFH), one of the agents was controlled by the VFH algorithm [87], and the other were human beings. Each experimental condition comprises seven different experiments (i.e. seven different trajectories), differing for the start and the goal chosen for the selected agent, the number of human subjects involved in the interaction, and their motion patterns.

The virtualized environment is constructed by processing movies collected from surveillance cameras of populated environments [92], obtaining a 2D arena where virtual agents reproduce the human motion captured in the movies. In the HO condition, the performance metrics are evaluated in the original arena, with reference to a randomly selected human being. In the GT and VFH conditions, a virtual agent is introduced in the arena and commanded to navigate through the existing virtual agents (corresponding to human beings) using the given trajectory planner.

3.2.5 Survey questionnaire, a-priori power analysis

Survey questionnaire

The proposed methodology is validated using a variation of the Turing test [93], which evaluates whether the robot behavior, controlled by the game-theoretical method, is comparable to or indistinguishable from human navigation patterns.

The variation of the Turing test consists of an online survey questionnaire composed of three main parts: (i) in the first part, the participant underwent a training phase to become familiar with the working environment (see Figure 3.3a-b); (ii) in the second part, the participant watched 21 videos reproducing the seven experiments for each of the three experimental conditions, where both the background and the agents are concealed –blue arrows over a gray background– (Figure 3.3c illustrates a frame of a single experiment); (iii) in the third and final part, the participant watched the same 21 videos (but in a different random order), where they were asked in addition to focus on a circled arrow (Figure 3.3d illustrates an example of a frame of a single experiment). The participant was unaware that the circled arrow targeted a random human agent in the HO experimental condition and the robotic agent in GT and VFH experimental conditions. We remark that the seven experiments used for the survey questionnaire are identical to those used to evaluate the performance metrics computed in the previous section.

The execution of each part entails answering specific questions. In the first part of the survey questionnaire, the participants were required to provide their gender, age, and level of experience in robotics field on a Likert scale [94] from 1 (no experience) to 5 (expert).

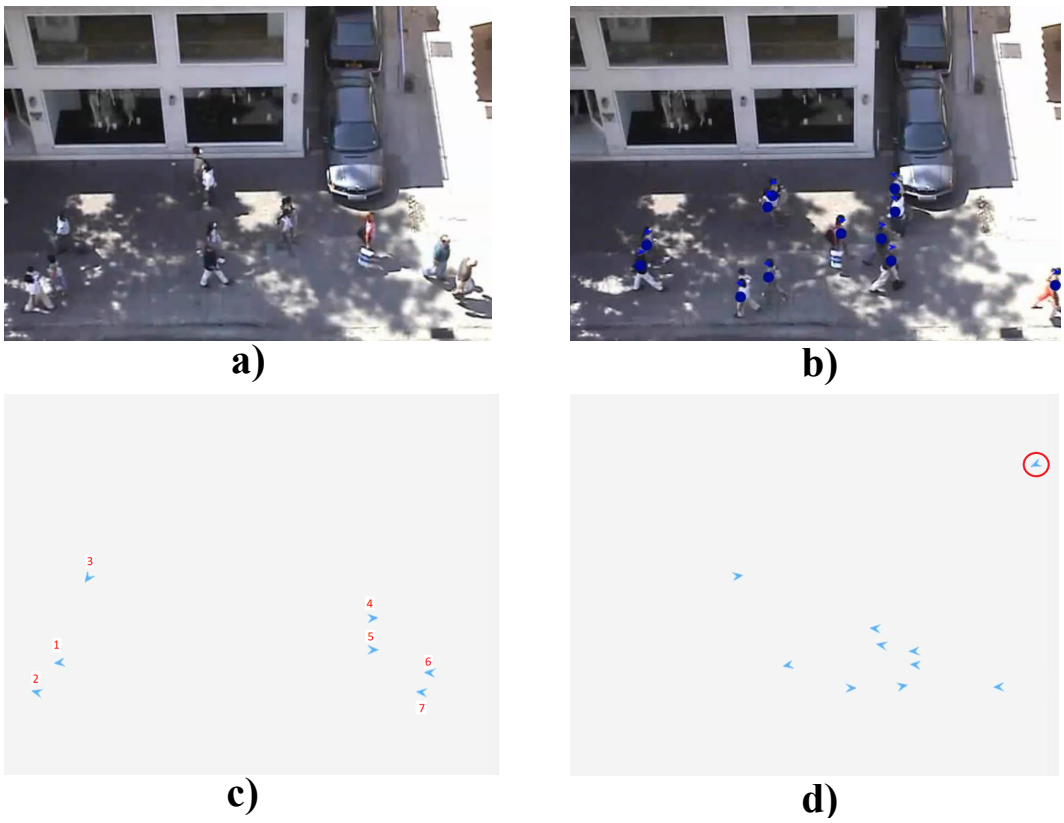


Fig. 3.3 Overview of the survey questionnaire.
a) Training part with open-source surveillance video [1]; **b)** Training part, intermediate scenario; **c)** Second part of the survey questionnaire, i.e. recognizing the motion of the "weird" arrow in the videos, if any; **d)** Third part of the survey questionnaire, i.e. follow the circled arrow.

During the training part, the participant is guided from a typical urban scenario of Figure 3.3**a** to the particular scenario used in the other parts of the test illustrated in Figure 3.3**c**. The intermediate scenario of Figure 3.3**b** is designed to gradually guide the participant to the final set-up.

In the testing scenario of Figure 3.3**c**, agents (pedestrians and robot) have been replaced with arrows and the urban environment has been removed to prevent the participant from focusing on the scenario, rather than on the movement of agents.

In the second part, the participant watches 21 videos randomly (about 15 seconds each) consisting of three different experimental conditions: 7 videos show an environment with only pedestrians (HO); other 7 videos a scenario with pedestrians and a robot controlled with an algorithm at the state-of-the-art (the Enhanced Vector Field Histogram [87]) (VFH); and the remaining 7 videos show a scenario with pedestrians and a robot controlled with the proposed algorithm (GT). In all experimental conditions, robot trajectories are re-planned with a frequency of 2 Hz.

To assess the level of social acceptance of our game-theoretical trajectories, in the second part (following habituation), we asked the participants to say if they perceived "weirdness" in the motion observed in the videos, and then to indicate which is the perceived "weird" arrow, if any, as shown in Figure 3.3**c**.

In the third part, participants were requested to determine whether the circled arrow was a human or not. Then, participants were asked to rate the naturalness of the motion of the circled arrow on a Likert scale [94] defined in a range from 1 (completely unnatural) to 5 (completely natural).

All videos used in the survey questionnaire are generated from an open-access dataset [92].

The test takes about 20 minutes to be completed properly. The test has three *rules*: *(i)* the participant cannot pause the video; *(ii)* the participant can watch videos only once; *(iii)* the participant should complete the test without interruptions or distractions.

A-priori power analysis

Preliminary, we conducted an a-priori power analysis to estimate the number of participants required to provide acceptable and significant statistical results [95].

To this aim, we used the free software G*Power [96]. First, we identified our case analysis as a non-parametric study, since non-parametric statistical tests make no constraints and prerequisites on the data distributions [97]. Then, we assumed that the data collected after the a-priori study would be analyzed via the non-parametric Kruskal-Wallis test because our *independent* variables have more than two independent groups (HO, GT, and VFH) and our *dependent* variables (the rating of the weirdness motion, human-likeness, and naturalness of movement) are ordinal.

Based on [95], we computed the total sample size considering the ANOVA test [98], i.e., the parametric-equivalent test of the Kruskal–Wallis one and then multiplied the result by the corrective factor ARE, obtaining the equivalent sample size of the non-parametric Kruskal–Wallis test. The result of the a-priori analysis for our non-parametric test is about 152 volunteers, considering an alpha level equal to 5%, power of the study 80% and the three number of groups, corresponding to the three different experimental conditions. We recruited the participants using the Institutional mail of Politecnico di Torino and then we distributed an online questionnaire to students and university staff. Ultimately, we collected 691 responses, exceeding the sample size of 152.

3.2.6 Statistical analysis

Experimental data (both the generated robotics trajectories and the responses to the survey questionnaire) were preliminary assessed for normality distribution and homoscedasticity of variance (Levene's test). These analyses revealed that data violated the assumptions for parametric statistics. Thus, we decided to adopt a non-parametric test, i.e., Kruskal-Wallis [99].

First, the quality of the trajectories generated by the two algorithms and the HO condition was evaluated. We first addressed whether they differed in terms of variability of path length ratio, path regularity, and distance to the closest pedestrian through the Levene's tests [100]. Significance level was set at $p < 0.05$ [99] (for all statistical tests performed in this study), and paired post-hoc comparisons have been conducted – adopting a Bonferroni correction – when appropriate. Following these preliminary analyses, trajectory data have been analysed through non-parametric Kruskal-Wallis test.

Survey questionnaire data have also been analysed through Kruskal-Wallis test followed by Bonferroni post-hoc analyses [101]. These analyses were aimed at assessing whether study participants exhibited a differential appraisal of the different trajectories in terms of weirdness, human likeness, and naturalness. This statistical approach was adopted for all the questions in the second and third part of the survey questionnaire, except for the second question of the second part. In the latter, participants were asked to indicate the perceived "weird" arrow, if any. We posit that more weirdness should be perceived in agents driven by algorithms than in agents associated with human beings. For this reason, the answers expressed relative to the HO scenario were not considered, since all arrows corresponded to human beings and an indication of weirdness would not make sense to our research question. As a consequence, in this specific instance, only two experimental conditions had to be compared (GT and VFH) and, to this aim, we used the Mann-Whitney test [102, 103].

3.3 Results

In this section, the results of the analysis conducted on the trajectories of the 21 experiments (seven experiments for each of the three experimental conditions) are presented. Then, the results of the survey questionnaire are illustrated and commented.

3.3.1 Analysis of performance metrics

Three widely adopted parameters, deemed as important for socially navigating robots, were evaluated across the three experimental conditions: the Path Length Ratio (PLR), the Path Regularity (PR), and the Closest Pedestrian Distance (CPD) [104] (for further detail see Section 2.3).

The PLR is defined as the ratio between the length of the line-of-sight path between the initial and final point of a path and the actual path length between the same two points [104]. A higher path length ratio is usually preferred, since it indicates that an agent minimizes the length of the path to reach its goal. We computed the PLR for each experiment and we illustrate its average values across the three experimental conditions in Figure 3.4a. The results in Figure 3.4a suggest

that the HO scenario was characterized by the highest average PLR, followed by GT and VFH.

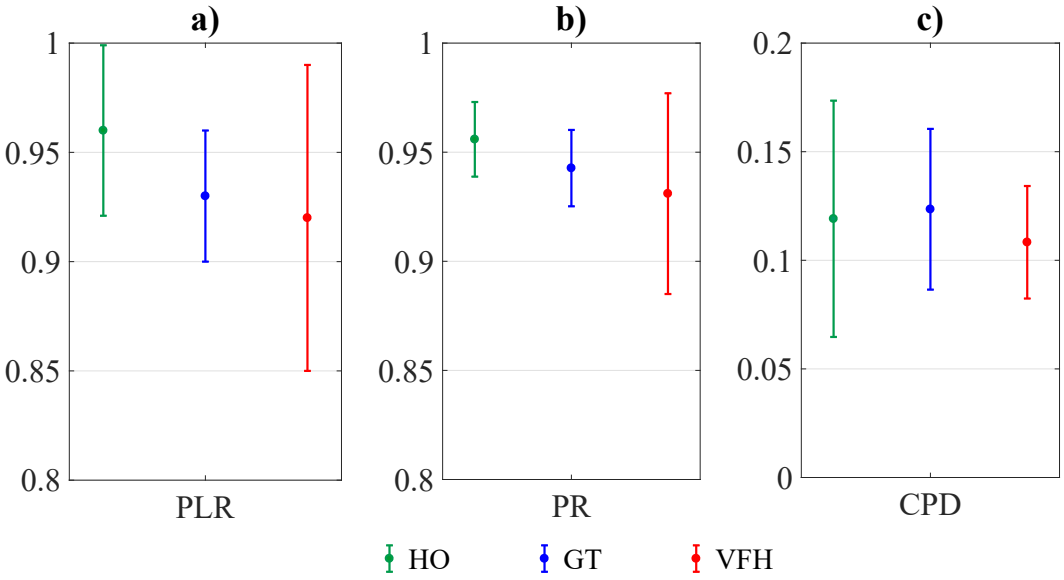


Fig. 3.4 The mean value and the standard deviation for each experimental condition considering each performance parameter are plotted in the figure. The three experimental conditions are: HO: video with humans only; GT: video with humans and a robot driven by a game-theoretical trajectory planner; VFH: video with humans and a robot driven by a vector field histogram algorithm. The performance metrics are: **a)** PLR (Path Length Ratio); **b)** PR (Path Regularity); **c)** CPD (Closest Pedestrian Distance).

The PR quantifies to what extent a path is similar to a straight line [104]. Following normalization, $PR = 1$ corresponds to a straight path from start to goal. Values of PR closer to one are preferable, since they are indicative of a smoother motion, without excessive changes of direction. In Figure 3.4b, the average PR for each experimental condition is illustrated, where the highest average value pertains to HO, followed by GT and VFH. These results appear in line with the tenet that humans tend to minimize their energy, thus avoiding sudden changes of orientation, and with the design principle of the VFH algorithm, which avoids obstacles only when the agent is close to them [87], entailing swift changes of orientation to get away from them.

The CPD is defined as the distance from the closest pedestrian, normalized with respect to the maximum length measurable during experiments, which is the diagonal of the experimental arena. Also for this parameter, the attainment of values closer to

one is desirable, as this implies a good tendency in staying clear from humans when following planned trajectories. Average values of CPD in the three experimental conditions are illustrated in Figure 3.4c, where the highest average value is related to GT, followed by HO and VFH. The reason for the latter is presumably due to the purely reactive design of the VFH algorithm. We posit that the intermediate ranking of HO with respect to CPD is due to the ability of humans to evaluate situations on a case-by-case basis.

While the rankings described above are suggestive of superior performance metrics attained by GT over VFH, the verification of the statistical significance of these comparisons is in order.

To preliminarily evaluate the quality of the trajectories generated by the two algorithms and the HO, we first addressed whether they differed in terms of inter-experiment variability of the three performance metrics through the Levene's test [100].

Hypothesis 1 (H_0) *The variance of the three performance metrics (PLR, PR, CPD) is statistically indistinguishable when computed over the three experimental conditions (HO, VFH, GT).*

Essentially, we evaluated the extent to which each algorithm generated trajectories that were similar to one another. Our analysis revealed that there exists a significant differential variability with respect to PR ($F_{2,18} = 3.75$, $p = 0.043$). Thus, we performed a post-hoc analysis that revealed much more variability in the VFH videos compared to HO and GT (VFH vs. HO: $p = 0.038$; VFH vs. GT: $p = 0.040$; HO vs. GT: $p = 0.97$). The inter-experiment variability within each experimental condition was indistinguishable concerning PLR ($F_{2,18} = 3.22$, $p = 0.064$) and CPD ($F_{2,18} = 2.31$, $p = 0.130$). These results indicate that, albeit indistinguishable in absolute values, the reproducibility and predictability of each experimental condition in terms of PR were much higher in HO and GT than in VFH scenario.

With the Leven's test described above, we have not only shown that the variances of the 3 experimental conditions in PLR and CPD are equal but we have also shown that for these two parameters the assumptions for doing the Kruskal-Wallis test are satisfied. Thus, in line with this consideration, the null hypothesis for the Kruskal-Wallis is defined as follows:

Number of participants	691
Gender	58% male and 42% female
Age	29.44±11.30
Experience with robotics	1.5±0.86

Table 3.1 Demographic characteristics and experience with robotics on a scale from 1 (minimum experience) to 5 (maximum experience) collected during the first part of the test.

Hypothesis 2 (H_0) *The two performance metrics (PLR, CPD) computed over the three experimental conditions (HO, VFH, GT) are statistically indistinguishable across experimental conditions.*

To this aim, Kruskal-Wallis analysis [105] was executed across the two performance metrics, revealing the non-achievement of significant statistical distinguishability ($\chi^2 = 2.5$, $p = 0.286$ for PLR; $\chi^2 = 0.36$, $p = 0.834$ for CPD). The reason behind such observations is strictly related to the consideration of only seven experiments for each experimental condition, with differential degree of variability, and thus characterized by a limited statistical power.

For completeness, we did an a posteriori power study to verify the limited statistical power, and what we found is that the statistical power considering only seven experiments per group is 6%, thus very limited.

3.3.2 Survey questionnaire

We collected 691 responses to the survey questionnaire, where participants were in majority men in their thirties with very little experience in robotics (Table 3.1). The age range of our sample goes from 18 to 78 years old.

A power analysis [95] indicated that the adequate statistical power was guaranteed with 152 participants. Since the number of participants largely exceeded the required sample size, we opted for a bootstrapping approach [106], in which we randomly sampled 152 observations from the complete pool of responses and iterated this process 100 times. Adopting this procedure, we kept the sample size to the appropriate number (thus reducing the odds of obtaining biologically irrelevant

findings [107]) and increased the generalizability of our findings by testing their robustness against repeated observations.

Experimental outcomes were analyzed with the Kruskal-Wallis test to statistically reject the H_0 hypothesis and understand if there exist differences among experimental conditions.

Our null hypothesis posits:

Hypothesis 3 (H_0) *All experimental conditions (HO, VFH, GT) are perceived by participants as indistinguishable.*

In the analysis of the results of the second part, in accordance with our expectations, the VFH condition was characterized by the highest level of weirdness compared to HO and GT conditions, which were, in turn, indistinguishable from one another (Kruskal–Wallis test $\chi^2 = 107 \pm 13.5$ and $p < 10^{-17}$ for all bootstrapping iterations; post-hoc analysis: for HO-VFH $p < 10^{-10}$ for all bootstrapping iterations, for GT-VFH $p < 10^{-14}$ for all bootstrapping iterations, for GT-HO $p > 0.05$ for 88 bootstrapping iterations out of 100, but the remaining has $p > 0.01$).

Figure 3.5a illustrates the mean rank (in light of the bootstrapping procedure) in "weirdness" of motion (WM) along with its standard deviation.

Notably, GT and HO are indistinguishable from one another, while VFH is significantly different from GT and HO. Specifically, while VFH was considered "weird" in the majority of instances (61%), GT was considered "weird" much less often than HO videos (33% and 37%, respectively) (See Figure 3.6).

We then asked the participants who detected weirdness in the videos to indicate which of the arrows exhibited such weirdness. We posit that more weirdness should be perceived in agents driven by algorithms than in agents associated with human beings. Our experiments indicated that the agent judged as weird was actually associated with a robot only in 16% of GT, while this proportion drastically increased to 47% of VFH (see Figure 3.6 patterned bars). This finding, combined with the Mann-Whitney test ($U = 4(10^3) \pm 489$, $p < 10^{-20}$ for all the bootstrapping iteration considering the whole bootstrapping analysis), supports the view that the trajectories generated by GT are perceived as much more natural than those generated by VFH. Additionally, it suggests that the motion of the robot controlled by GT is perceived as more human-like than the one generated by VFH.

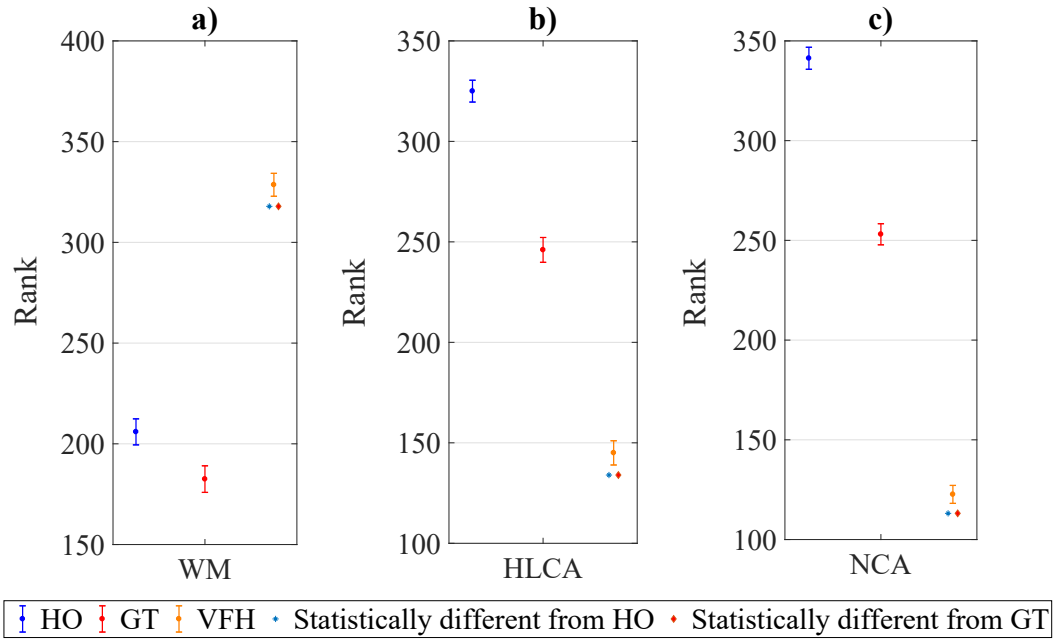


Fig. 3.5 Summary of the post-hoc Kruskal-Wallis test of the survey questionnaire. The mean rank of each group is plotted for each part of the test with the corresponding standard deviation considering all bootstrapping iterations. **a)** Second part of the test in which the attention of the participant is not focused on one arrow in particular. WM: weirdness motion. **b)** - **c)** Third part of the test in which the participant is focused on the circled arrow. HLCA: human-likeness of the circled arrow; NCA: naturalness of the circled arrow. HO: video with humans only; GT: video with humans and a robot driven by a game-theoretical trajectory planner; VFH: video with humans and a robot driven by a Vector Field Histogram algorithm. The blue asterisk highlights the statistical difference from HO, instead the red diamond highlights the statistical difference from GT.

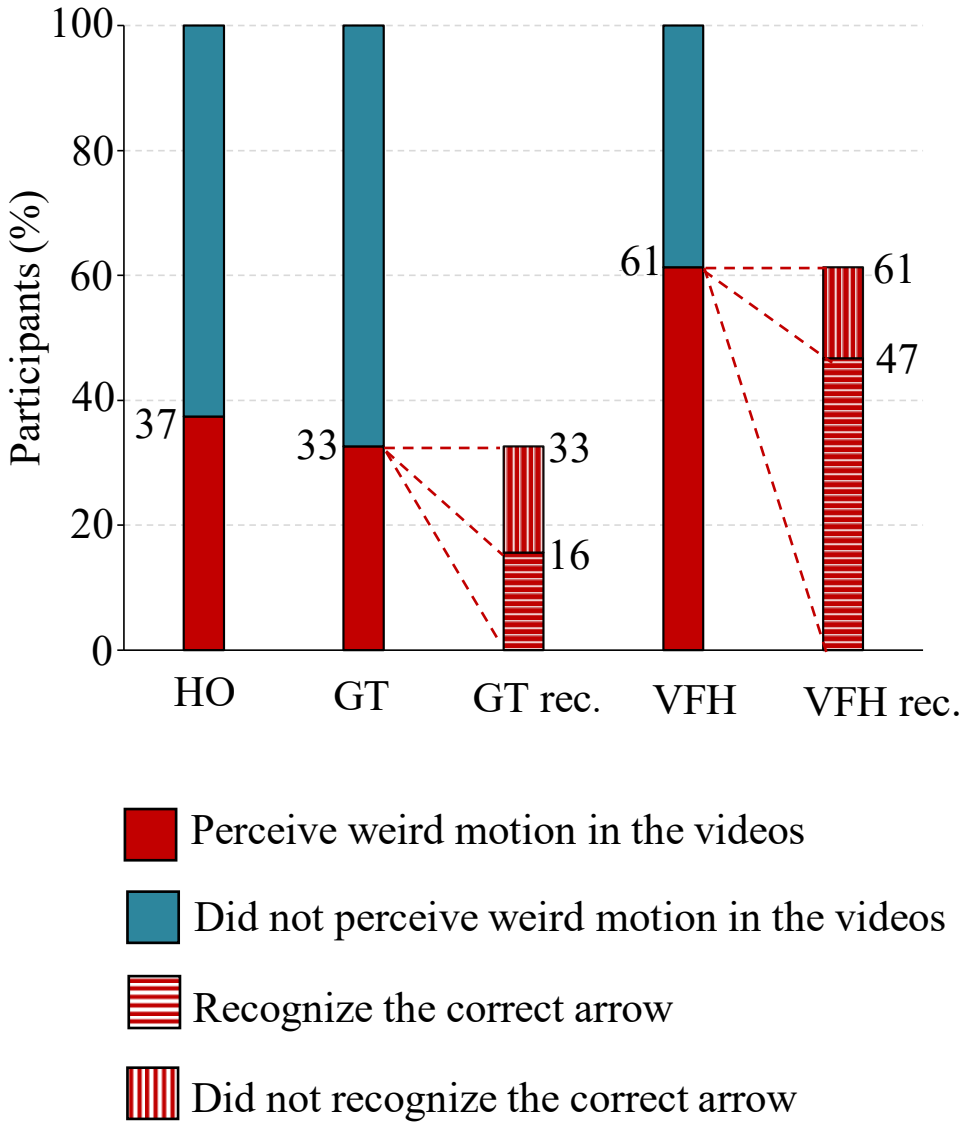


Fig. 3.6 Result of the second part of the survey questionnaire considering the participants that perceive a "weird" motion in the 3 groups of experimental conditions and recognize the correct arrow (the robot) in the populated environment (GT rec., VFH rec.). HO: video with humans only; GT: video with humans and a robot driven by a game-theoretical trajectory planner; VFH: video with humans and a robot driven by a vector field histogram algorithm.

In the third part, we further delved into the subjective rating of the three motion patterns by asking participants to focus on the motion of a circled target agent and evaluate whether such motion corresponds to a human or not (human likeness), along with its degree of naturalness on a Likert scale from one (minimum naturalness) to five (maximum naturalness). When focusing on the qualitative measurements of the human likeness, we observed that VFH-related arrows were considered much less human-like (41.11%) than both GT (64.59%) and HO (80.31%). Thus, as illustrated in Figure 3.5b, VFH is judged as the least human-like ($\chi^2 = 142.55 \pm 15.12$, $p < 10^{-22}$ for all Kruskal-Wallis bootstrapping iterations; post-hoc analysis: $p < 10^{-5}$ VFH-GT, $p < 10^{-30}$ VFH-HO) which is consistent with the previous part of the test, where VFH is perceived as generating the "weirdest" motion. Additionally, GT-related arrows were considered significantly less human-like compared to HO ($p < 10^{-4}$ post-hoc analysis GT-HO).

Figures 3.5c and 3.7 illustrate the results related to the naturalness of the circled arrow. The figure 3.7 shows the result about the average naturalness of motion of the circled arrow on a Likert scale from 1 (minimum naturalness) to 5 (maximum naturalness), computed over the 100 iterations of the bootstrapping procedure.

According to our expectations, HO exhibits the highest mean degree of naturalness (4) with a standard deviation of 0.04, closely followed by GT (3.5) with a standard deviation of 0.04, whereas a larger gap separates VFH (2.6) with a standard deviation of 0.05.

Importantly, although significantly different from HO, GT values exceeded three. This may indirectly suggest that while HO videos were deemed natural, also GT videos may have been regarded as human-like. Yet, this preposition is currently speculative whereby the intermediate value (three) was not marked with the anchor natural. Therefore, future studies are needed to precisely detail the individual appraisal of the naturalness of the GT trajectory.

3.4 Discussion

The main goal of our study was to design a navigation system for autonomous robots moving through populated environments, characterized by a high degree of acceptability by humans. Specifically, in light of the increasing use of autonomous

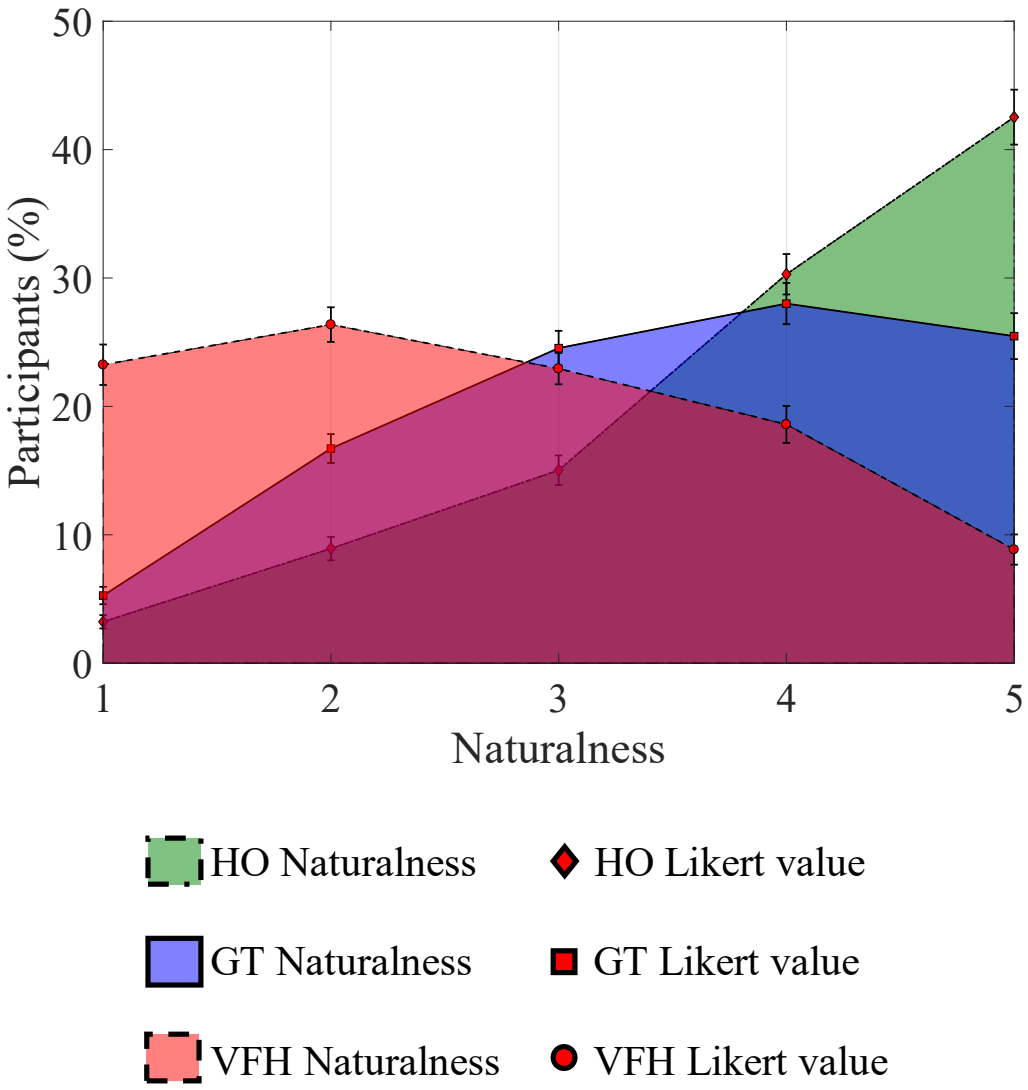


Fig. 3.7 Result of the third part of the survey questionnaire. The participant assigns a degree of naturalness on a Likert scale from 1 (minimum naturalness) to 5 (maximum naturalness) considering the circled arrow in the 3 experimental conditions. The red points on the figure show the average naturalness of each rate of the Likert scale considering 100 iterations with the bootstrapping approach, and the error bars represent the standard deviation. HO: the circled arrow is human; GT: the circled arrow is a robot driven by the game-theoretical trajectory planner; VFH: the circled arrow is driven by the Enhanced Vector Field Histogram algorithm.

robots in real life, we tested whether a navigation system designed through the principles of game theory would generate indistinguishable trajectories from those walked by human beings. To this aim, we first leveraged game theory to develop a model capable of predicting the intention of motion of humans in populated environments, and then, based on this model, we devised a trajectory planning algorithm for a mobile robot. Finally, to assess the social acceptance of the generated robotic trajectories, we conducted a survey questionnaire on a statistically robust group of volunteers using a variation of the Turing test.

For greater completeness and toward even more robust outcomes, before analyzing the results collected from the test, we also analyzed the geometrical features of the robotic trajectories, generated in the three experimental conditions (HO, GT, and VFH), selecting three performance metrics from the state of the art (PLR, PR, CPD). The ranking obtained through this analysis (HO, GT, VFH) is consistent with the results obtained through the Turing test, except for the closest pedestrian distance (CPD), in which the trajectories generated by our planner (GT) exhibit higher values of the parameter than those measured in environments populated by humans only (HO). We hypothesize that this exception is due to the fact that our model guarantees by design a minimum safe distance to pedestrians to prevent collisions and to ensure, in any case, a comfortable action space. On the other hand, humans on the walk are more flexible in this respect, and evaluate circumstances on a case basis. While the outcome of the Turing test is consistent with the analysis of the performance metrics of the trajectories, the statistical analysis (Kruskal-Wallis) executed on the latter shows that this finding is not statistically significant. To explain this non-statistically significant result, we point out that the statistical analysis was conducted on only seven experiments per group, with differential degree of variability, and thus characterized by a limited statistical power.

Moreover, to preliminary evaluate the quality of the trajectories, we conducted a systematic analysis (Levene's test) to assess the degree of variability of the different scenarios. In other terms, we evaluated the extent to which each algorithm generated videos that were similar to one another. This analysis revealed that there exists a significant variability with respect to the path regularity (PR), whereby the videos with the robot controlled by the VFH are the most variable, compared to the HO and GT experimental conditions. This finding suggests that the VFH algorithm is less predictable (i.e., it provides less regular results) than our algorithm and a real pedestrian.

The variant of the Turing test comprises a first part that functions as a training phase. The second part comprises two consecutive phases. The first phase is devoted to compare the social acceptability of trajectories generated by either our game-theoretical algorithm (GT) or a state of the art algorithm (VFH) against a reference experimental condition, a complex social environment populated by humans only (HO). To this aim, participants were asked to say if they perceived weirdness in HO, GT, or VFH experimental conditions. The statistical test confirms that the perceived weirdness in trajectories in which only human subjects are involved is statistically indistinguishable from trajectories where the GT-controlled robot and human subjects coexist. Conversely, the videos in which the trajectories are generated by the VFH algorithm are perceived with a remarkably higher degree of overall weirdness compared with either HO or GT scenarios.

In the second phase of the second part of the test, participants were asked to indicate which is the perceived "weird" arrow, if any. In this regard, we observed that the trajectories generated by the VFH algorithm were more frequently recognized as "weird" than those generated by our GT algorithm.

In the third part of the test, participants were requested to focus on a circled arrow (a human in the HO experimental condition, a robot in GT and VFH ones), and were asked to evaluate whether or not the motion of the circled arrow corresponded to human recordings, and then rate their degree of naturalness. We observed that, while the arrow in VFH scenario was perceived as not human-like, the arrow controlled through GT was considered human-like, albeit not as human-like as the one rated in the HO experimental condition. We believe that this result is related to the fact that, in this part of the test, participants were asked to focus on one arrow only, thus being biased toward detecting an artificial behavior. The same ranking between the three experimental conditions (HO, GT, and VFH) resulted from the analysis of the naturalness of motion of the circled arrow. Indeed, HO has the highest degree of naturalness, closely followed by our GT trajectory planner, and then by the VFH planner.

We can conclude that, if participants are not guided to focus on a particular arrow, they would not distinguish much difference between a real human and a robot controlled through our game-theoretical framework and, therefore, the generated trajectory is a good candidate for social acceptance. This implies that our trajectory planning algorithm would help programming robots to blend well in populated

environments, and, hence, to be perceived as more friendly, collaborative, and non-hostile.

Our findings are consistent with other studies in the literature, such as [74], where a different game-theoretical planner is perceived almost as human-like as human recordings. However, in [74], the authors created a human-like motion planner for mobile robots, still maintaining a simplified framework that does not comprise, for example, human groups, obstacle avoidance performed by humans, and the human desire to keep a safe personal space around them [29]. Moreover, their tests only comprise simplified scenarios: a first test with either only humans, or only robots; a second test in which the participant, based on virtual reality, interacts with an agent who can move as a human or a robot. In our study, we went one step further in modeling (including the personal space, the group recognition, and the human-obstacle interaction) but also in the design of the variation of the Turing test (considering a real case scenario in which a robot moves in a human populated environment). Nevertheless, it is hard to make extensive comparisons with other approaches, as the literature on variants of the Turing tests for assessing social acceptability of a robot agent is scant.

Notably, the literature reports three main methods to evaluate the human-likeness and the social acceptance of robot navigation: (i) definition of social rules or performance metrics and, then, assessment of the adherence of the robot motion planner to these principles [108–110]; (ii) comparison between simulated trajectories and observed pedestrian behavior [111]; (iii) questionnaire based on a variation of the Turing test [112, 74]. The main limitation of the first two methods is that they do not consider how humans perceive the robot. However, these methods can be applied to evaluate, as a preliminary test, some features of the generated trajectories. Indeed, our analysis of the performance metrics of the generated trajectories falls within the first methodology, whereas the second methodology has been used as a validation criterion for our game-theoretical model of pedestrian motion.

Hence, toward our aims, we deemed the Turing as an effective means to study the human-likeness and the social acceptability of the generated trajectories.

Unlike the Kretzschmar's [112] and Turnwald's [74] tests, where volunteers watched videos in which the totality of agents moved either in an artificial way or as real pedestrians, our questionnaire changes completely such a perspective. In fact, our test videos reproduce a true use case scenario of the algorithm (an

environment populated by people with a single robot moving within), where the real nature of agents is masked and made uniform to eliminate any participants' bias. Moreover, unlike Kretzschmar's test [112], where the Turing test is executed only on 10 participants, we performed an a priori power analysis to infer the correct sample size to obtain statistically significant results. Due to the largely superior size of collected data than the outcome of the power analysis, we carried out a 100-iteration bootstrap, always getting consistent results across iterations, highlighting the robustness of our results and further corroborating our hypothesis.

When interpreting the results of our study, we should also acknowledge the limitations of the **model** and of the **test** design. Regarding the former, our **model** does not take into account the uncertainties that arise from the interaction with the external world. Importantly, the stochasticity of human behavior is not explicitly modeled, although this is implicitly accounted for through tuning model parameters identified from real trajectory data, extracted from surveillance cameras. A range of simplifying assumptions were in order to handle the computational complexity of the algorithm. The main one resides in the discrete nature of our model, whereby each agent can choose between a fixed number of motion directions –an indispensable trade-off between predictive accuracy and computational effort. Moreover, the designed human motion model has been devised to operate with a limited number of pedestrians: its computational complexity may be difficult to manage if the number of agents increases to more than a dozen. The pedestrian model used in this study only considers people's goal-directed and collision-avoiding behaviors, while ignoring other social activities that humans may perform in a pedestrian urban scenario, such as waiting for a bus or wandering without a clear direction. Thus, any pedestrian behavior that is not contemplated by our model breaks the assumptions under which our system works. In addition, our method does not allow customization of trajectories. For example, the prediction of a trajectory walked by an elderly person may be coincident with that of a child.

The main limitations of the **test** design are two: (i) the choice of the navigation algorithm chosen for comparison (VFH); (ii) the use of pre-recorded pedestrian trajectories for the design of the videos. These trajectories do not account for the potential influence of the robot on human motions.

Regarding the first limitation, ideally, more than one algorithm should have been selected in order to mitigate algorithm-induced biases. However, since the execution

of the Turing test already took about 20 minutes to the average participant, we prefer to limit our comparison to only one algorithm at the state of the art, in order to avoid increasing the time of the experiment for each participant, mitigate attention biases and, in the end, achieve robust results.

Regarding the second limitation, while neglecting pedestrian reactions to the robot's movement might be considered a limitation, this design choice aligns with the core objective of this test: assessing whether participants recognize the robot controlled by the GT algorithm in a human-populated environment and whether they perceive its generated trajectory as human-like. Incorporating pedestrian reactions to the robot's movement in the videos would have required a human motion model. However, no perfect human motion model exists. Any such model would inevitably incorporate assumptions about the nature of pedestrian walking behaviors, introducing biases in aspects such as preferred speeds, accelerations, and acceptable personal space. Thus, including a model that simulated human reactivity would have introduced additional "artificial agents" into the videos, potentially introducing significant biases for the ultimate objective of the study. To avoid this problem, we opted to place the robot in an arena with pre-recorded human trajectories.

We acknowledge that real-world experiments are still necessary to provide a complete picture of social navigation performance. However, we believe that our simplified approach provides a valuable first step in evaluating the potential of our algorithm to generate human-like robot trajectories in populated environments.

Our work can be extended in several directions. To manage and predict the motion of big crowds, mean-field games could be adopted [113]. We remark, however, that crowded and populated scenarios are different in many aspects, and the deployment of a robot in the two scenarios would cover totally different application fields.

The lack of customization in the inference of trajectories by our model can be mitigated by combining our approach with learning strategies as in [114], encompassing variegate behaviors across the experimental scenario. In fact, adding variability to the pedestrian model might allow for a more accurate prediction of human motion pattern and should allow the robot to better adapt to the needs of the human with whom it is interacting. For example, if a robot recognizes a person who has difficulties in walking, the robot should be able to predict their movement and possibly reduce its speed. Moreover, it would be interesting to understand and assess the quality of our generated trajectories considering not only social acceptability but

also the comfort [54] feeling of participants, for instance by creating a real shared environment with humans and a robot.

Chapter 4

Game theoretical social force model (GTSFM) for human prediction

To operate efficiently in human-populated environments, robots must be able to recognize and consider mutual influences between humans, particularly in navigation scenarios where mutual avoidance maneuvers are necessary. Furthermore, achieving social acceptability for robots requires them to exhibit human-like behavioral patterns, as individuals are more inclined to trust and interact with robots perceived as similar to themselves. For these reasons, it is of paramount importance to develop navigation models that incorporate such a human motion feature. To achieve this goal, in this chapter, we propose a novel human motion model in which we encapsulate a game-theoretic decision-making mechanism within the established framework of the social-force model (SFM). The proposed model is able to predict the decision of multiple agents as they interact with each other during navigation, and to plan their optimal path toward reaching their goal. The model is compared against the standard SFM through the use of two state-of-the-art performance metrics (Path Length Ratio and Path Regularity). Our numerical results, performed in different scenarios with an increasing number of agents, indicate that the proposed model outperforms the classical SFM in all scenarios. This model will be used in Chapter 5 to design a trajectory planning for a mobile robot in a shared space with humans.

This chapter is organized as follows. Section 4.1 provides an outline of the methodological approach employed to model the human motion in social robot navigation, as well as a review of the state of the art in human behavior modeling. In

Section 4.1.2, we preliminarily introduce the SFM. Then, Section 4.2 presents our novel human motion model. Finally, Section 4.3 presents and discusses the results of our simulation experiments.

For a comprehensive review of the key concepts in socially-aware navigation, please refer to Chapter 2.

4.1 Overview of the approach and main contributions

In this chapter, we propose a novel model of human motion, building on the well-established social-force model (SFM) [20] and utilizing a game-theoretic formalism to integrate human decision-making within navigation tasks. Our model is designed to provide a reliable tool for comprehending human intentions in populated environments, with the ultimate goal of utilizing this information to control the motion of robots in populated environments, as described later in Chapter 5.

In literature, multiple approaches have been proposed to model human motion. Among others, these methods include reactive models [20, 68], learning methods [22–25], and game-theoretic approaches [69, 115, 61, 62, 116].

The SFM [20] is one of the pioneering human motion models, where humans are modeled as particles subject to a set of forces, which includes an attractive force, which guides the particles toward a desired goal, and one or multiple repulsive forces, which ensure collision avoidance among pedestrians and obstacles. A similar approach is proposed in [68], where a human motion model results from the combination of attractive and repulsive potential fields. These models, which rely on the laws of Newtonian mechanics, allow for implementations with low computational complexity. For this reason, they found successful real-world applications, particularly in the context of multi-agent navigation [117].

Despite powerful, these modeling approaches have some key limitations. They are purely reactive, whereby they overlook the fact that humans possess the ability to make predictions, manage conflicts, and cooperate with others while navigating [118]. This simplicity may limit their performance in terms of optimality of the planned path and of social acceptability [74].

Toward addressing this limitation, recent efforts have obtained promising results using learning-based approaches to predict human motion [22–25]. However,

learning-based approaches often require large training data sets. Moreover, they suffer in handling changes in scenarios, as they may require to be completely re-trained upon such changes. Additionally, the black-box nature of learning-based approaches limits the interpretability of the selected actions.

Alternative approaches rely on game theory. Despite the well-known ability of the game theory to model different aspects of human behavior, only a few efforts have attempted to incorporate it into human motion models. Hoogendoorn et al. [115] combined optimal control and differential games to describe human motion as an optimal feedback control trying to reach goals by minimizing the cost of navigation. The solution of the game stems from the solution of an optimal control problem. The main limitation of this work is the assumption that pedestrians react only to pedestrians that are directly in front of them, while other pedestrians are neglected. Rahamati et al [116] combines game theory and learning approaches modeling the interaction between pedestrians and the nearest agent exploiting game theory. However, such approach relies on the assumption that human motion is only influenced by the nearest neighbor, and more complex interactions are neglected. Turnwald [69] uses two types of non-cooperative games and posits that the attainment of the Nash equilibrium in a static game describes human planning and interactions more accurately than the prediction of the behavior of single individuals. The interesting conclusions arising from this work are affected by two important limitations: first, the test is limited to two players, whereas settings with multiple pedestrians are neglected. Second, the authors assume that the decision process is executed only once, overlooking the possibility of changing a decision during navigation, a situation that normally happens in human navigation [85].

Here, in line with the model developed in Chapter 3, we use non-cooperative game theory [56] to model the navigation behavior of multiple humans in populated environments. Specifically, we devise a game whose action set is a finite set of possible trajectories for each pedestrian. However, unlike Chapter 3, these possible trajectories are generated using the well-established and computationally-efficient SFM over a fixed-time horizon. Then, each pedestrian selects their action toward minimizing a cost function that takes into account their willingness to reach their goal, the regularity of their trajectory, and their willingness to avoid interactions with other individuals within their personal space. The general structure of our *game-theoretical social force model* (GTSFM) is illustrated in Figure 4.1.

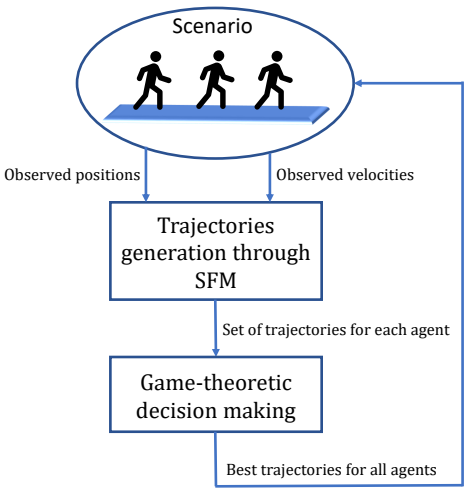


Fig. 4.1 Conceptual structure of the game-theoretic social force model (GTSFM).

Our model extends the current state-of-the-art along many directions. First, our model keeps the computational efficiency of the SFM, while using it to estimate multiple trajectories over a fixed-time horizon for each pedestrian. Second, we incorporate game theory into our approach to determine the optimal trajectory for each individual, taking into account the mutual influences between humans. Third, unlike prior works that focus just on binary interactions [115, 116], in our model each pedestrian interacts by playing a game with all other pedestrians present in the scene. Furthermore, in contrast to [69], in our model the decision-making process is calculated at every time step rather than just once, mimicking the sequential nature of human decision-making processes.

After formalizing our model, we run a campaign of numerical simulations to test its performance. Specifically, we compare performance metrics (viz. Path Length Ratio and Path Regularity) of the trajectories generated using our model with those generated using the standard SFM in different scenarios of populated environments. The results of our numerical validation suggest that the GTSFM always exhibit superior performance than the SFM across all scenarios. Overall, there is a significant improvement in performance (up to about 30% compared to the standard SFM). Such an improvement becomes less evident when the environment becomes too crowded, e.g., in settings with 20 pedestrians in a 10×16m environment, suggesting that our method should be preferred in populated, but not densely crowded, environments.

4.1.1 Notation

We gather here some notational conventions used throughout this chapter. The sets of natural, real, real nonnegative, and strictly positive real numbers are denoted by $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$, respectively. We use roman font to denote scalar quantities ($x \in \mathbb{R}$) and bold font to denote vectors in the plane ($\mathbf{x} \in \mathbb{R}^2$). Given a vector $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$, $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2}$ denotes its Euclidean norm.

4.1.2 Social-Force Model

The SFM was introduced by D. Helbing and P. Molnar in [20] to reproduce the motion of pedestrians in an environment. Over the years, numerous scholars have suggested various enhancements to refine and improve the Helbing model [119–121]. In this study, we are inspired by the works of Helbing [20] and Ferrer [120].

The SFM considers a set $\mathscr{N} = \{1, \ldots, n\}$ of $n \in \mathbb{N}$ pedestrians, who move in a continuous planar space $X \in \mathbb{R}^2$. Each pedestrian $i \in \mathscr{N}$ is characterized by a goal $\mathbf{p}_i^{\text{goal}} \in \mathbb{R}^2$, that is, the final position that the pedestrian wants to reach, and two time-varying vectors that determine their current state: their position $\mathbf{p}_i(t) \in \mathbb{R}^2$ and their velocity $\mathbf{v}_i(t) \in \mathbb{R}^2$. Position and velocity evolve in continuous time, $t \in \mathbb{R}_{\geq 0}$. The motion of each pedestrian is governed by a set of forces. In particular, the model includes an attractive force $\mathbf{F}_i^{\text{goal}}$, which drives the pedestrian toward their goal, a set of interaction forces of the form $\mathbf{F}_{ij}^{\text{int}}$, which makes pedestrian i to avoid colliding with pedestrian j, and a repulsive force related to the proximity of the pedestrian to an obstacle or wall $\mathbf{F}_i^{\text{obs}}$.

The motion of the generic ith pedestrian is obtained by representing the pedestrian as a particle complying with the laws of Newtonian mechanics. The equations of motion of pedestrian i are given by the following system of ordinary differential equations discretized with a time step Δ_t

$$\begin{aligned} \mathbf{v}_i(t+\Delta_t) &= \mathbf{v}_i(t) + \Delta_t \left[\mathbf{F}_i^{\text{goal}}(t) + \sum_{j \in \mathscr{N} \setminus \{i\}} \mathbf{F}_{ij}^{\text{int}}(t) + \mathbf{F}_i^{\text{obs}}(t) \right] \\ \mathbf{p}_i(t+\Delta_t) &= \mathbf{p}_i(t) + \Delta_t \mathbf{v}_i(t), \end{aligned} \tag{4.1}$$

where in the first equation we have assumed, without any loss in generality, that forces are re-scaled so that all pedestrians have unitary mass. In the following, we detail the three types of forces present in Equation 4.1.

Attractive force: Pedestrian i is attracted to their goal $\mathbf{p}_i^{\text{goal}}$ by means of the driving force $\mathbf{F}_i^{\text{goal}}(t)$. Specifically, fixed a scalar parameter $v_i^d > 0$, which is the desired velocity of the individual toward their goal, we define

$$\mathbf{F}_i^{\text{goal}}(t) = \frac{v_i^d \widehat{\mathbf{e}}_i(t) - \mathbf{v}_i(t)}{\alpha_i}, \tag{4.2}$$

where $\widehat{\mathbf{e}}_i(t) = \frac{\mathbf{p}_i^{\text{goal}} - \mathbf{p}_i(t)}{\left\|\mathbf{p}_i^{\text{goal}} - \mathbf{p}_i(t)\right\|}$ is the desired direction (i.e., the normalized vector that points toward the pedestrian's goal) and $\alpha_i \in \mathbb{R}_{>0}$ is a parameter that captures the relaxation time. The latter regulates the rate of change of the pedestrian –the smaller values mimic a stronger tendency to decisively move toward the goal.

Interaction forces: The motion of pedestrian i is influenced by the interaction with other pedestrians. In fact, pedestrians tend to feel uncomfortable when unknown pedestrians get close to their personal space and consequently try to avoid this situation [19]. In the SFM, the interaction of pedestrian i with pedestrian j is modeled with a repulsive force F_{ij}^{int} (defined for all $j \in \mathscr{N} \setminus \{i\}$). We denote by $r_i, r_j > 0$ the radii of the personal spaces of pedestrians i and j, respectively (i.e., the space that a pedestrian tends to keep void of other pedestrians), and by $d_{ij}(t) := ||\mathbf{p}_i(t) - \mathbf{p}_j(t)||$ the distance between the two pedestrians. Then, the interaction force is defined as

$$\mathbf{F}_{ij}^{\text{int}}(t) = A_i \exp\left\{\frac{r_i + r_j - d_{ij}(t)}{B_i}\right\} F_{\text{fov}}(t)\widehat{\mathbf{n}_{ij}}(t), \tag{4.3}$$

where the constant parameters $A_i, B_i \in \mathbb{R}_{>0}$ regulate the strength and range of the interaction force for pedestrian i, respectively. $F_{\text{fov}}(t) \in [0,1]$ is a (time-varying) scaling factor, associated with the pedestrian's field of view, detailed in the following, and $\widehat{\mathbf{n}_{ij}}(t) = \frac{\mathbf{p}_i(t) - \mathbf{p}_j(t)}{\left\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\right\|}$ is the direction between the two pedestrians.

Since pedestrians have a limited field of view when walking and are affected mainly by the objects within their field of view, the repulsive force is scaled

by an anisotropic factor $F_{\text{fov}}(t) \in [0,1]$ depending on the bearing $\gamma_{ij}(t)$ of pedestrian j, measured from pedestrian i[1].

Such a factor is defined as

$$F_{\text{fov}}(t) = \lambda + (1-\lambda)\frac{1+\cos(\gamma_{ij}(t))}{2}, \tag{4.4}$$

where $\lambda \in [0,1]$ is the strength of the anisotropic behavior: the closer λ is to one, the smaller is the impact of a limited field of view.

Repulsive forces from obstacles/walls: During navigation, a pedestrian instinctively maintains a specific distance from the edges of buildings, walls, and obstacles. The discomfort increases as the individual approaches a border, as greater attention is required to avert the risk of injury, such as accidental contact with a wall. Consequently, a border of a generic obstacle elicits a repulsive effect, which is characterized by the following expression:

$$\mathbf{F}_i^{obs}(t) = A_{obs} \exp\left\{\frac{r_i + r_o - d_{i,obs}(t)}{B_{obs}}\right\} F_{i,obs}^{fov}(t)\widehat{\mathbf{n}}_{i,obs}(t), \tag{4.5}$$

where the constant parameters $A_{obs}, B_{obs} \in \mathbb{R}_{>0}$ regulate the strength and range of the interaction force for pedestrian i, respectively. $r_i, r_o > 0$ the radius of the personal spaces of pedestrians i and a value to model a safety distance from the nearest point of the nearest obstacle, respectively. $d_{i,obs}(t) := ||\mathbf{p}_i(t) - \mathbf{p}_o(t)||$ is the distance between the pedestrian i and the obstacle. As discussed for the repulsive force between pedestrians (Equation 4.3), also the repulsive force for obstacle is scaled by a corresponding anisotropic factor $F_{i,obs}^{fov}(t)$ defined as follows:

$$F_{i,obs}^{fov}(t) = \lambda + (1-\lambda)\frac{1+\cos\gamma_{i,obs}(t)}{2} \tag{4.6}$$

where $\lambda \in [0,1]$ is the strength of the anisotropic behavior: the closer λ is to one, the smaller is the impact of a limited field of view and $\gamma_{i,obs}(t)$ is the bearing angle of the considered obstacle measured from pedestrian i. Moreover, in the definition of the Equation 4.5, there is $\widehat{\mathbf{n}}_{i,obs}(t)$, i.e. direction

[1] More specifically, the bearing $\gamma_{ij}(t)$ is the angle between the direction of motion of agent i and the segment joining the positions of agent i and agent j.

Table 4.1 Model variables and parameters.

Symbol	Meaning
n	number of pedestrians
$X \in \mathbb{R}^2$	planar space
$\mathbf{p}_i(t) \in \mathbb{R}^2$	position of pedestrian i at time t
$\mathbf{v}_i(t) \in \mathbb{R}^2$	velocity of pedestrian i at time t
$\mathbf{p}_i^{\text{goal}} \in \mathbb{R}^2$	position of the goal of pedestrian i
$\mathbf{F}_i^{\text{goal}} \in \mathbb{R}^2$	attractive force for pedestrian i at time t
$\mathbf{F}_i^{obs}(t) \in \mathbb{R}^2$	repulsive force of the closest obstacle on pedestrian i at time t
$\mathbf{F}_{ij}^{\text{int}}(t) \in \mathbb{R}^2$	interaction force of pedestrian j on i at time t
$r_i \in \mathbb{R}_{>0}$	radius of the personal space of pedestrian i
$r_j \in \mathbb{R}_{>0}$	radius of the personal space of pedestrian j
$v_i^d \in \mathbb{R}_{>0}$	desired velocity (in modulus) of pedestrian i
$\alpha_i \in \mathbb{R}_{>0}$	relaxation time of pedestrian i
$A_i \in \mathbb{R}_{>0}$	strength of interaction force for pedestrian i
$A_{obs} \in \mathbb{R}_{>0}$	strength of repulsive force from obstacle i
$B_i \in \mathbb{R}_{>0}$	range of interaction force for pedestrian i
$B_{obs} \in \mathbb{R}_{>0}$	range of repulsive force from obstacle
$\gamma_{ij}(t) \in \mathbb{R}$	bearing of pedestrian j measured by pedestrian i
$F_{\text{fov}} \in [0,1]$	anisotropic factor in pedestrian interaction
$F_{i,obs}^{fov} \in [0,1]$	anisotropic factor for pedestrian-object interaction
$\gamma_{i,obs}(t) \in \mathbb{R}$	bearing of the nearest obstacle measured by pedestrian i
$\lambda \in [0,1]$	strength of anisotropic behavior

between the pedestrian i and the nearest point of the nearest obstacle defined as $\widehat{\mathbf{n}}_{i,obs}(t) = \frac{\mathbf{p}_i(t)-\mathbf{p}_o(t)}{\|\mathbf{p}_i(t)-\mathbf{p}_o(t)\|}$.

All the parameters of the SFM are summarized in Table 4.1.

4.2 Methodology

4.2.1 Game-Theoretic Social-Force Model (GTSFM)

In this section, we present our model of human motion, which encapsulates a game-theoretic decision process within the SFM. For this reason, in the following, we will

refer to our model as the game-theoretic social-force model (GTSFM). To simplify the presentation we will refer to pedestrians as *agents*.

In our game-theoretical model, all pedestrians are considered *rational agents* operating in a 2-dimensional dynamic environment that is populated with other agents. Furthermore, all agents possess *common knowledge*, that is, they have access to the same set of available actions to reach their final goal and the behavior of other pedestrians while walking. This assumption is particularly relevant in modeling human behavior, as individuals tend to learn these skills through experience in their daily lives [74].

4.2.2 Game-theoretic formalization

Here, we present our model for pedestrian motion, which is a non-cooperative, static, perfect information, and finite game with a finite number of agents. In our game, each agent aims to achieve its own goal *individually*.

In the proposed GTSFM, the n agents $\mathcal{N} = \{1, \dots, n\}$ move according to the SFM and use a game-theoretic mechanism to regulate the strength A_i of their interaction forces with other agents. This mechanism models a rational decision process to establish the relative importance of two social forces, namely, the intent to reach the goal and the willingness to avoid other agents entering one's own space.

Specifically, agent i can select the value of parameter A_i from a finite set of possible values $\mathscr{A}_i$. A cost function $J(a)$ associates to each action $a \in \mathscr{A}_i$ its cost. In a general setting, such cost can be associated with different characteristics of the motion, such as path length [88] or energy consumption [89]. The objective of each agent is to minimize their cost function. Here, we posit that the optimal behavior for humans is the convergence to a Nash equilibrium, a situation where no agent has an incentive to unilaterally change their action without the others changing theirs [69].

In the following, we provide the details on the cost function defined for the GTSFM, we illustrate the method used to compute the Nash equilibria of the game, and we finally summarize our motion algorithm.

4.2.3 Cost function

We assume that agents have perfect information, that is, they have information about the current actions of other agents. This assumption is realistic, considering the inherent human characteristic of interpreting others' behaviour and predicting their motion [69]. Furthermore, human decision-making can be susceptible to biases and may not always account for potential interactions effectively. To address this inherent uncertainty and incorporate also the reactive nature of humans, the SFM logic is employed.

Using the SFM and the perfect information assumption, each agent can compute the trajectory associated with each value of the parameter A_i. In particular, each trajectory is computed by using Equation (4.1) over a fixed-time horizon $(T_{\text{prev}}\Delta_t)$, where T_{prev} is the number of time-steps and Δ_t is the duration of each time-step. To enhance clarity and without sacrificing generalizability, here and henceforth, we assume a unitary discrete-time step, i.e. Δ_t=1.

Hence, to each $a \in \mathscr{A}_i$, we associate the corresponding trajectory for agent i, denoted as τ_i^a.

Finally, to evaluate the cost of each generated trajectory (τ_i^a) for the ith pedestrian, we define the cost function $J(\tau_i^a)$ as the sum of three contributions:

$$J(\tau_i^a) = \Phi_{\text{goal}}(\tau_i^a) + \Phi_{\text{smooth}}(\tau_i^a) + \Phi_{\text{int}}(\tau_i^a). \tag{4.7}$$

Such a cost function is built on the cost function proposed in Chapter 3, enriched with a term that accounts for the willingness to engage in social interaction ($\Phi_{\text{int}}(\tau_i^a)$).

Unlike Chapter 3, where interaction with other agents was implemented as a hard constraint, here we incorporate it as a soft constraint within the cost function. In particular, in the previous chapter, the hard constraint was useful because when the algorithm generated candidate trajectories for each agent, it adopted a simple combination of the agent's possible actions. This approach resulted in a significant number of infeasible candidate trajectories, such as trajectories that came too close to other agents. Thus, these infeasible candidate trajectories necessitated post-processing via a hard constraint for removal. In contrast, the current chapter leverages the SFM to generate the candidate trajectories for each agent thus achieving two key advantages: (i) the candidate trajectories for each agent already incorporate

interaction with other agents. This eliminates the need for the post-processing step; (ii) all candidate trajectories for each agent are feasible, requiring only the selection of the optimal one through the evaluation of the designed cost function (Equation 4.7).

Details on the three summands of Equation 4.7 are given in what follows.

Path length The first term is defined as

$$\Phi_{\text{goal}}(\tau_i^a) = \sum_{k=1}^{T_{\text{prev}}} \|\mathbf{p}_i(t+k) - \mathbf{p}_i^{\text{goal}}\|, \tag{4.8}$$

and accounts for the path length for agent i. Hence, minimizing such a term captures the goal-oriented attitude of the agent.

Path regularity The second term is defined as

$$\Phi_{\text{smooth}}(\tau_i^a) = \sum_{k=1}^{T_{\text{prev}}} |\theta_i(t+k) - \theta_i(t+(k-1))|, \tag{4.9}$$

where $\theta_i(t+k)$, $\theta_i(t+(k-1))$ are the orientation (angle) of the agent at time-step $(t+k)$ and $(t+(k-1))$, respectively. Such a term penalizes excessive rotations, thus promoting smooth trajectories. In fact, during navigation, humans tend to avoid too many changes of orientation to minimize their energy consumption [89].

Interaction with other pedestrians The third term is defined as

$$\Phi_{\text{int}}(\tau_i^a) = \sum_{j \in \mathcal{N} \setminus \{i\}} \sum_{k=1}^{T_{\text{prev}}} \frac{\rho}{\|\mathbf{p}_i(t+k) - \mathbf{p}_j(t+k)\|}, \tag{4.10}$$

where $\rho \in \mathbb{R}_{\geq 0}$ is a (constant) weighting factor, and the denominator in Equation (4.10) is the distance between the agent position $\mathbf{p}_i(t+k)$ and the other pedestrians $\mathbf{p}_j(t+k)$ at time-step $(t+k)$. Hence, by minimizing this term, humans tend to maintain a safe distance from others. The weighting factor ρ determines the relative weight of this term with respect to the other two in the cost function and, ultimately, the radius of the agents' personal space.

4.2.4 Numerical computation of Nash equilibria

Following [69], we assume that agents during navigation tend to converge to a Nash equilibrium. This refers to a situation where no agent has an incentive to unilaterally change its action without the others changing theirs.

However, the existence and uniqueness of a Nash equilibrium cannot be guaranteed in our setup, and its analytical characterization is unattainable. Therefore, as employed in the algorithm presented in Chapter 3, we use numerical methods to compute an approximate Nash equilibrium, specifically through the *sequential best response* approach [91]. The details of this approach are given in Section 3.2.1.

The game-theoretical model for human motion presented above has been implemented following the logic shown in Algorithm 4. During its main loop, Algorithm 4 invokes the game-theoretical strategy selection, which logic is described in Algorithm 5.

Algorithm 4: Main algorithm performed by agent $ag \in N$.

```
p_ag ← InitializeAgentPosition
τ ← FirstEstimation [straight paths for all agents]
while p_ag ≠ p_goal do
    y ← 1 [iteration index]
    τ^y ← τ
    while τ^y ≠ τ^(y−1) do
        y ← y + 1
        foreach agent i (included ag) ∈ N do
            (τ_i^*)^y ← StrategySelection (i, τ)
    p_ag ← UpdateAgentPosition
```

4.3 Results of the human motion based on GTSFM

In this section, we perform a campaign of numerical simulations to test the performance of the proposed GTSFM. Specifically, we consider 200 simulations for each scenario, with an increasing number of pedestrians involved (viz. $n \in \{4,6,8,10,12,15,20\}$). In each scenario, the n pedestrians are positioned in a rectangular domain of size 16×10 meters (equal in all scenarios). Initial positions,

Algorithm 5: Computation of the actions with the SFM and strategy selection for an agent

StrategySelection$(i, \boldsymbol{\tau})$
  foreach *action a* **do**
    while $k < T_{\text{prev}}$ **do**
      $m_i \frac{\mathrm{d}v_i}{\mathrm{d}t} = \mathbf{F}_i^{\text{goal}}(t+k) + \sum_{j \in \mathcal{N} \setminus \{i\}} \mathbf{F}_{ij}^{\text{int}}(a, t+k)$
      $v_i(t+k) = v_i(t+(k-1)) + \frac{\mathrm{d}v_i}{\mathrm{d}t}$
      $\mathbf{p}_i(t+k) = \mathbf{p}_i(t+(k-1)) + v_i(t+k)\Delta t$
      $k = k+1$
    $(J(\tau_i^a)) \leftarrow \text{CostComputation}(\tau_i^a)$
  $(\tau_i^*)^y \leftarrow \text{MinimizationCost}(J(\boldsymbol{\tau}_i))$
  return $(\tau_i^*)^y$

Number of agents (n)	4	6	8	10	12	15	20
Improvement *PLR*	31.61%	30.30%	30.87%	30.66%	25.62%	20.78%	9.59%
Improvement *PR*	19.51%	20.52%	18.33%	24.01%	24.68%	23.59%	19.50%

Table 4.2 Results of our numerical performance comparison between the SFM and the GTSFM in terms of the improvement of the performance metrics of the GTSFM with respect to the SFM.

as well as individuals' goals, are sampled uniformly at random in the rectangular domain, with the additional constraint that initial positions and goals of different pedestrians are sufficiently separated, that is, preventing them from being one within the personal space of another.

To compare the performance of the established SFM with the performance of the proposed GTSFM, we consider two different performance metrics: the Path Length Ratio (*PLR*) and the Path Regularity (*PR*) [82]. Given that this study involves simulations with n agents, we adapt the performance metrics outlined in Section 2.3 for n agents.

The *PLR* is computed as

$$PLR := \frac{1}{n} \sum_{i=1}^{n} \frac{\|\mathbf{p}_i(0) - \mathbf{p}_i(T_{\text{goal}})\|}{\sum_{k=1}^{T_{\text{goal}}} \|\mathbf{p}_i(k) - \mathbf{p}_i(k-1)\|}, \tag{4.11}$$

where T_{goal} is the number of time steps needed for the agents to reach their goals, considering a unitary duration of each time step for simplicity. A higher *PLR* value

is generally desirable, as it suggests that the pedestrian takes the shortest possible path to reach their destination.

The *PR* is computed as:

$$PR = 1 - \frac{\sum_{i=1}^{n} \sum_{k=1}^{T_{\text{goal}}} |\theta_i(k) - \theta_i(k-1)|}{PI_{\max}}, \tag{4.12}$$

where $\theta_i(k)$, is the orientation (angle) of the agent i at time k, and the denominator is a normalizing factor defined as

$$PI_{\max} = \max \sum_{i=1}^{n} \sum_{k=1}^{T_{\text{goal}}} |\theta_i(k) - \theta_i(k-1)|, \tag{4.13}$$

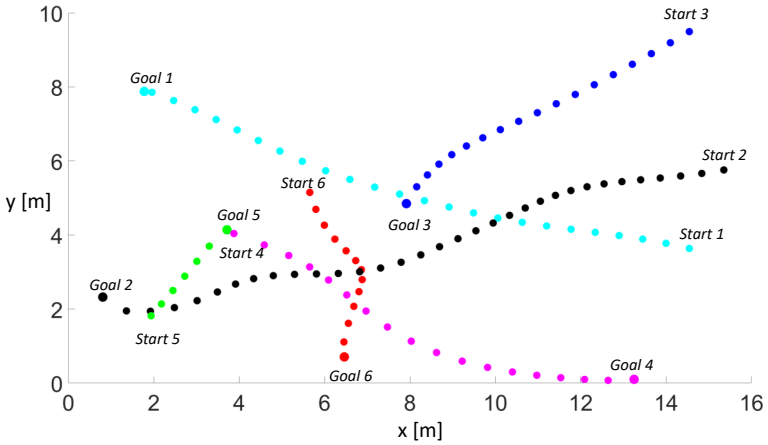
where the maximization should be intended over all the realizations in the sample set.

The index *PR* measures the (normalized) pedestrian rotation during their navigation. Such a normalized index takes values in $[0,1]$: $PR = 1$ indicates a straight path from the starting point to the goal, while lower values of *PR* denote more rotation during the execution of the trajectory. It is desirable to have *PR* values closer to 1 since they indicate a smoother motion with fewer sudden changes in direction.

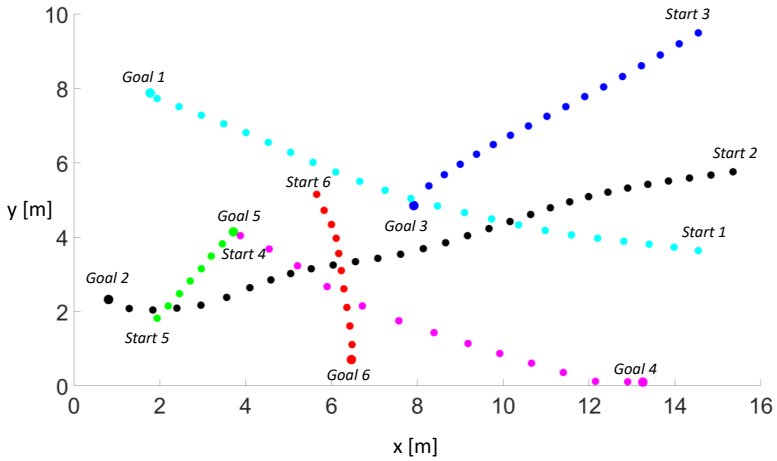
In our simulations we fix the remaining parameters to $r_i = 1.2$m, $v_i^d = 0.6$m/s, $A_i = [0.05, 0.06, 0.07, 0.08, 0.09, 0.1]$, $B_i = 1$m, $\lambda = 0.6$, $T_{\text{prev}} = 5$ time-steps of duration $\Delta_t = 0.9$s each. A representative comparison between two simulated trajectories is reported in Figure 4.2, where we can immediately visualize how the proposed GTSFM generates collision-free trajectories that are smoother than the one generated with SFM —see, e.g., the red trajectory, which is sensibly straighter than the one of the SFM, and reaches the goal in 10 steps instead of 12.

In view of the stochasticity of the simulation setting, for each of the 7 scenarios, we perform 200 independent simulations in which we randomize initial position and goals, and we average the results obtained adopting a Monte Carlo approach.

The results of our Monte Carlo numerical simulations are reported in Table 4.2. In particular, in the table, we report the results in terms of the decrease in the gap with respect to the value of the performance metrics in the ideal (but not necessarily feasible) scenario of a straight path, where $PRL = PR = 1$, when using a GTSFM with respect to the SFM. Results are averaged over the 200 simulations. Hence,



(a) SFM



(b) GTSFM

Fig. 4.2 Comparison between the simulated trajectories of $n = 6$ agents in our simulation setting, for (a) the SFM and (b) the GTSFM. The objective of each pedestrian is to arrive at the goal indicated in the figure.

the first entry of the table means that using the GTSFM instead of the SFM would decrease the gap of path length ratio with respect to the optimal (not necessarily feasible) trajectories on average by 31.61%, in environments with $n = 4$ agents.

Our numerical validation confirms that GTSFM shows a significant improvement for all scenarios. In particular, it decreased the gap with respect to the ideal straight path up to 31.61% over the SFM when considering the *PLR*. In terms of the *PR*, we observed a significant improvement with respect to the SFM, spanning from 18.33% up to 24.68%, indicating that the trajectories generated by our model steer clear of unnecessary rotations and instead prioritize more direct paths. In light of this finding, it can be concluded that our human motion model can sensibly improve the route of a standard SFM while avoiding potential agents, ultimately leading to trajectories that closely approximate the shortest possible path.

Similar performance metrics are observed in all the simulations. However, as the environment becomes more crowded, the improvement decreases. In fact, in the simulation setting with 20 agents, the performance of the GTSFM is still higher, but the improvement between the two methods is less pronounced (especially, the *PLR* improves only by 9.59%). We conjecture that the reason for such a performance decrease is that, in highly crowded environments, the presence of many pedestrians makes difficult to anticipate the movement of all of them. This is a common limitation of human motion models, as different kinds of models are employed to reproduce crowds, e.g., multiscale models [122].

4.3.1 Discussion

In this chapter, we presented a novel game-theoretic approach to predict human motion in a populated environment. Our approach, termed game-theoretic social-force model (GTSFM), relies on the encapsulation of a game-theoretic decision-making mechanism within the well-established SFM [20]. The proposed model inherits the computational efficiency of the SFM, while it incorporates the capability of predicting others' motion, typical of game-theoretic approaches. To test the efficiency of the proposed approach, we have compared the performance of the GTSFM with the one of a standard SFM in different scenarios, using two state-of-the-art performance metrics [82]. Numerical results showed that the proposed

GTSFM outperforms the standard SFM in all simulation scenarios, both in terms of the path length ratio and the path regularity.

The presented preliminary results on GTSFM can be employed to design a trajectory planner for a real mobile robot sharing space with humans. In fact, in the following chapters (Chapters 5 and Chapters 6), GTSFM is used to create an algorithm for a mobile robot to make it socially acceptable. In this context, we evaluate the quality of the generated trajectory considering *quantitative* performance metrics and *qualitative* measurements such as the naturalness of motion and the human feeling of comfort [54].

Chapter 5

GTSFM trajectory planning to enhance robot social acceptance through *quantitative* analysis

This chapter presents a novel social navigation algorithm for mobile robots operating in human-interactive environments. The algorithm prioritizes both human safety and comfort while efficiently guiding the robot towards its designated goal location. To reach this objective, the human motion model (developed in Chapter 4) informs the design of a robotic trajectory planner, whereby the robot tends to mimic human behavior during motion. Notably, in this chapter, we improve the GTSFM model by adding the real-time estimation of the parameters for the SFM through a differential evolution algorithm. To *quantitatively* assess the GTSFM algorithm, we conducted a comparative analysis with two state-of-the-art algorithms using four established performance metrics commonly employed in the field. These metrics specifically evaluated the novel algorithm's efficiency in achieving its goals and the smoothness and comfort of the generated trajectories compared to existing state-of-the-art approaches.

This chapter is organized as follows. Section 5.1 provides an outline of the approach used in this study to design and evaluate the socially aware navigation algorithm established based on the findings of Chapter 4. In Section 5.1.1, the novelty of our approach is presented. In Section 5.2, the methodology used is described. Then, in Section 5.3, the tools used to perform the *quantitative* analysis are presented.

Finally, Section 5.4 presents and discusses the quantitative results of our simulation experiments.

5.1 Overview of the approach

In Chapter 4, we used the non-cooperative game theory [56] combined with the well-known social force model to model (SFM) the navigation behavior of multiple humans in populated environments. Here, the human motion model informs the design of a robotic trajectory planner, whereby the robot tends to mimic human behavior during motion and interaction in a populated environment.

Similar to Chapter 3, the phenomenon of anthropomorphism is also leveraged here (see Section 2.1 for further details).

By incorporating aspects of human movement into robotic trajectories, we aim to amplify perceived anthropomorphism, ultimately leading to greater human acceptance [33].

Specifically, we devise a game whose action set is a finite set of possible trajectories for each agent, generated using the well-established and computationally efficient SFM over a fixed-time horizon. Then, each agent selects the action toward minimizing a cost function. To find different SFM parameters sets and generate different trajectories for each agent, a Differential Evolution (DE) algorithm [3] is incorporated in the GTSFM framework. While the DE algorithm offers powerful parameter estimation capabilities, its high computational demand limits its application in real-time scenarios. This study addresses this limitation by proposing a neural network that, when properly trained, can mimic the behaviour of the DE algorithm, enabling efficient and real-time parameter estimation. The general structure of our *game-theoretical social force model* (GTSFM) is illustrated in Figure 5.1.

Following the formalization of our algorithm with the new real-time parameters estimation logic, a *quantitative* evaluation of its performance is conducted. The *quantitative* evaluation is done involving robot-human interaction in simulated scenarios. Focusing on navigation efficiency, trajectory smoothness and comfort, we assess performance against two state-of-the-art algorithms using four performance metrics. Notably, the results demonstrate that our algorithm generates significantly

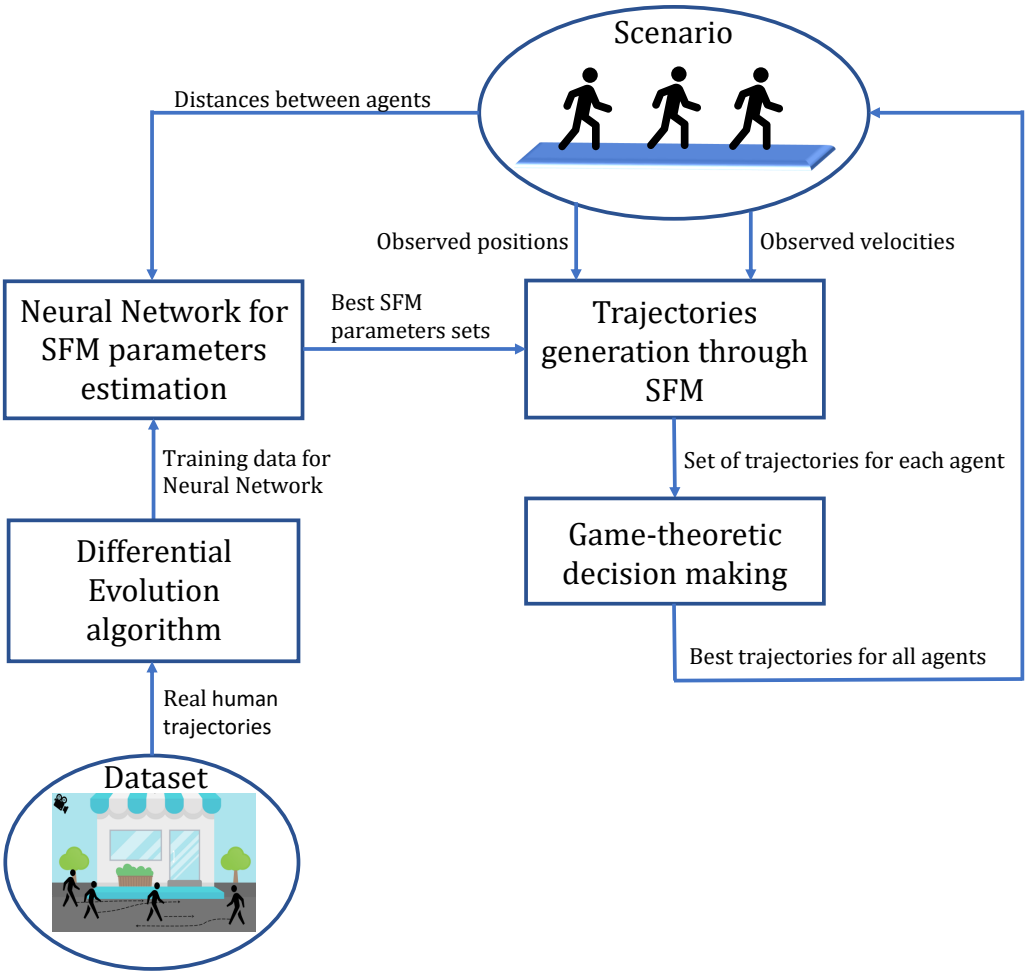


Fig. 5.1 Conceptual framework of GTSFM with real-time parameters estimation.

smoother paths compared to the other two state-of-the-art algorithms, potentially leading to more natural robot motion.

Furthermore, the evaluation reveals that our approach outperforms the standard SFM model across all considered performance metrics, highlighting its potential for improved navigation efficiency and trajectory quality.

To complement the *quantitative* analysis, a *qualitative* evaluation using a real robot is conducted in Chapter 6. This evaluation aimed to assess how our algorithm is perceived in comparison to two other state-of-the-art algorithms. However, the *qualitative* analysis does not identify any algorithm that shows significant superiority over the others. This lack of distinction can be attributed to unaccounted factors. The robot's appearance could have obscured the distinction between the algorithms.

Additionally, the limited velocity of the real robot may have limited the range of conditions tested among the different algorithms and, consequently, obscured the distinction between the algorithms.

5.1.1 Our contributions

This thesis proposes a novel trajectory planning approach by combining the SFM with the game theory logic (GTSFM). This combined approach offers significant advantages over existing methods within the model-based and learning-based categories (for details on each category, please refer to the Section 2.2).

With respect to the former, our model extends the current state-of-the-art in many directions. First, our model keeps the computational efficiency of the SFM, while using it to estimate multiple trajectories over a fixed-time horizon for each pedestrian. Second, we incorporate game theory into our approach to determine the optimal trajectory for each individual, taking into account the mutual influences between humans. Third, while similar approaches have been investigated in different contexts, such as in a road-shared space with pedestrians and cars [123], the integration of the SFM and game theory for socially-aware navigation appears to be novel. Fourth, unlike prior works that focus just on binary interactions [115, 116], in our model, each agent interacts by playing a game with all other agents present in the scene. Furthermore, in contrast to [74], in our model the decision-making process is calculated at every time step rather than just once, mimicking the sequential nature of human decision-making processes. Finally, our approach can generate human-like trajectories by using a differential evolution algorithm, which tries to approximate real human trajectories as closely as possible.

On the other hand, our proposed approach offers several advantages over learning-based methods. First, it does not require training on a specific scenario. This makes it more generalizable to unseen situations, which is crucial for real-world applications. Second, our model provides more interpretable results. This is because it is based on a physics-based model of the human motion, rather than a black-box learning model. This interpretability is essential for understanding the reasons behind the robot's decisions and for limiting potentially unsafe behaviors.

Since the GTSFM algorithm is built upon the SFM, the GTSFM inherits both its strengths and limitations. A key strength of the SFM is the ability to accurately

describe the dynamics of pedestrian movement and various navigational phenomena, enhancing its utility in diverse research settings [117]. On the other hand, the main limitation of the SFM is related to the parameter dependency of the model, which results in two main problems: the calibration and homogeneity of the parameters. About the former, the model is extremely sensitive to even slight changes in the parameter values (see Figure 5.2). Notably, the SFM parameters may vary considerably depending on the context in which the model is applied. Thus, finding the optimal parameters for a given navigation scenario often requires an extensive and time-consuming trial-and-error process.

While the model excels in capturing general pedestrian behavior, it currently faces limitations in representing individual differences. Real-world navigation reveals diverse responses to stimuli due to personal feelings and motivations, translating to an ideal parameter value for each agent. However, the model's difficulty in calibrating parameters often necessitates using a single homogenous set for all agents. This assumption compromises the model's descriptive ability, as it cannot fully capture the heterogeneity observed in human navigation.

Thus, to address the two problems presented above, we use a DE algorithm to estimate the SFM parameters that best approximate real human trajectories.

This thesis employs the DE due to its advantageous features over the state of the art algorithms, such as the Evolutionary algorithms (EA) [3]. Unlike many EAs, DE eliminates the need for encoding real-valued parameters into bit strings, leading to reduced computational complexity and implementation effort. Additionally, its implementation requires minimal lines of code, making it accessible to researchers across diverse fields. Furthermore, DE's performance depends on tuning just a few well-studied control parameters (population size, mutation factor, and crossover rate), facilitating efficient parameter tuning.

Due to these characteristics, the DE has become one of the most popular optimization algorithms used in research [3]. Notably, its impressive adaptability allows the DE algorithm to tackle a wide range of optimization problems across diverse fields, such as aircraft control [124], localisation problem of a robot [125], and planning robot trajectories [126].

In our approach, DE is used to estimate the SFM parameters that best approximate the trajectories of a real-human trajectory dataset (Thör [2]). While the DE algorithm offers robust optimization, its computational demands limit its suitability

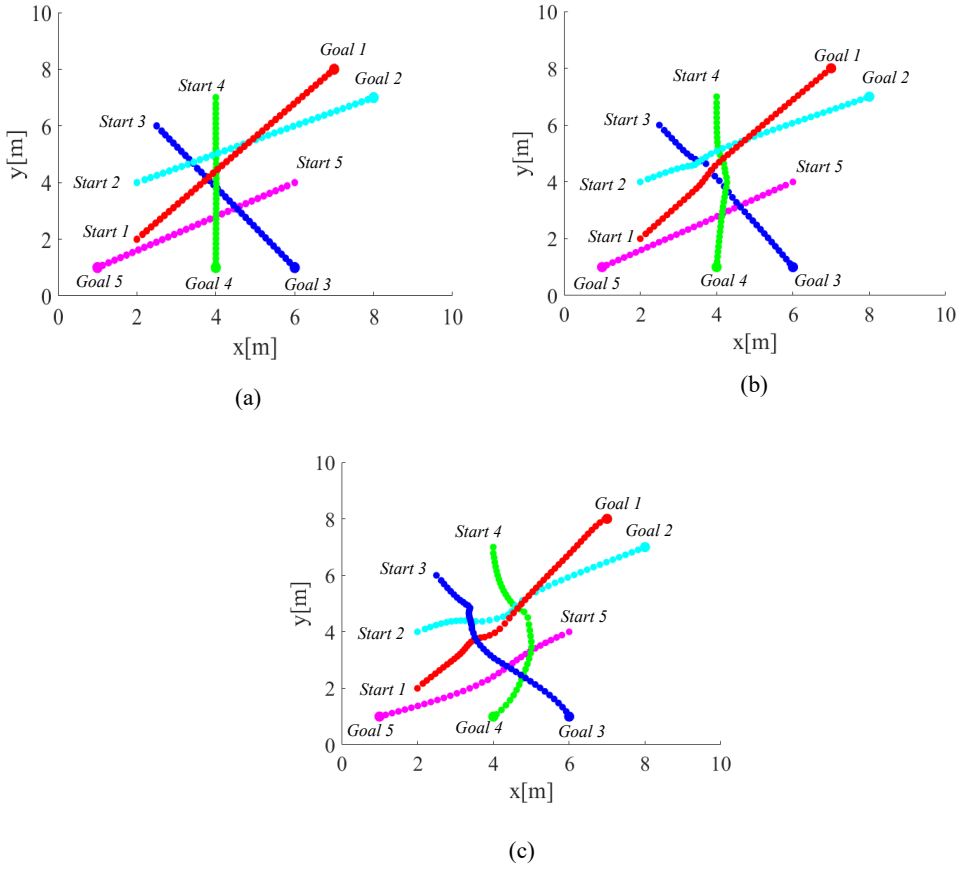


Fig. 5.2 Simulations with 5 agents controlled by SFM with different sets of parameters: (a) $A_i = 0.2, B_i = 0.1, r_i = r_j = 0.1$; (b) $A_i = 0.45, B_i = 0.3, r_i = r_j = 0.4$; (c) $A_i = 1, B_i = 0.7, r_i = r_j = 0.7$. For illustrative purposes, we introduced variability by adjusting only the pedestrian interaction parameters while assuming that all pedestrians share identical values for the desired speed (v_i^d), relaxation time (α_i), and anisotropic strength (λ). Nevertheless, it's important to note that this simplification does not always reflect the real-world situation accurately.

for real-time parameter estimation. This study tackles this limitation by proposing a neural network architecture. Following appropriate training on data representing the outcomes of the DE algorithm, this neural network can mimic the behavior of the DE algorithm, enabling efficient and real-time parameter estimation.

5.2 Overview of GTSFM with real-time parameters estimation

In this chapter, a GTSFM algorithm with real-time parameter estimation is adopted. In Figure 5.1, the general structure of the algorithm is presented. Specifically, we devise a game whose action set is a finite set of possible trajectories for each agent, generated using the well-established and computationally efficient SFM over a fixed-time horizon. Then, each agent selects their action toward minimizing a cost function that takes into account their willingness to reach their goal, the regularity of their trajectory, and their willingness to avoid interactions with other individuals within their personal space.

To find the best SFM sets of parameters approximating human behaviour and generate different trajectories for each agent, a DE algorithm [3] is incorporated in the GTSFM framework, as shown in Figure 5.1.

However, the excessive time needed to perform the parameter estimation makes the DE algorithm inadequate for real-time applications. Thus, to overcome this issue, the DE algorithm is approximated by a neural network. The latter is trained with supervised data, leveraging a labelled dataset created by the DE algorithm. The labelled dataset associates specific features of the scenario, such as the distance between agents, with the optimal parameters for the SFM.

To compute the optimal parameters for the training dataset, the DE algorithm computes the best sets of parameters for the SFM that approximate real human trajectories of a public dataset (Thör [2]).

The proposed model will subsequently be employed to perform two distinct tasks: (i) model and predict the motion of pedestrians; (ii) design a robotic trajectory planner informed by the GTSFM model.

Regarding the details of the SFM logic, please refer to Section 4.1.2. In the following section, we overcome the game presented in the Section 4.2 where the actions are generated modifying the value of A_i. Precisely, in Section 5.2.1, the focus shifts from chosing only the A_i value to the selection of an entire parameters set, offering a more realistic approach. Then, the three additional blocks are presented: the dataset, the DE algorithm, and the neural network. Each of these blocks is explained individually and comprehensively.

5.2.1 Game-theoretic formalization

Here, we present a game-theoretic methodology which is a non-cooperative, static, perfect information, and finite game with a finite number of agents. In our game, each agent aims to achieve its own goal *individually*.

In the proposed GTSFM, the n agents $\mathscr{N} = \{1, \ldots, n\}$ move according to the SFM and use a game-theoretic mechanism to find the best set of parameters to generate the best trajectory to reach the goal and avoid other agents.

Specifically, agent i can select the best set of parameters among a finite sets of parameters $\mathscr{S}_i$. We leverage the previously defined cost function $J(a)$ (see Section 4.2.3 for details) to associate a cost to each action $a \in \mathscr{S}_i$.

The objective of each agent is to minimize their cost function. Here, we posit that the optimal behavior for humans is the convergence to a Nash equilibrium, a situation where no agent has an incentive to unilaterally change their action without the others changing theirs [69]. For the method used to compute the Nash equilibria of the game please refer to Section 4.2.4.

We assume that agents have perfect information, that is, they have information about the current actions of other agents. This assumption is realistic, considering the inherent human characteristic of interpreting others' behavior and predicting their motion [69].

Using the SFM and the perfect information assumption, each agent can compute the trajectory associated with each parameters set. In particular, each trajectory is computed by using Equation (4.1) over a fixed-time horizon $(T_{\text{prev}}\Delta_t)$, where T_{prev} is the number of time-steps and Δ_t is the duration of each time-step. To enhance clarity

and without sacrificing generalizability, here and henceforth, we assume a unitary discrete-time step, i.e. Δ_t=1.

Hence, to each $a \in \mathscr{S}_i$, we associate the corresponding trajectory for agent i, denoted as τ_i^a.

Finally, to evaluate the cost of each generated trajectory (τ_i^a) for the ith pedestrian, we define the cost function $J(\tau_i^a)$ as the sum of three contributions:

$$J(\tau_i^a) = \Phi_{\text{goal}}(\tau_i^a) + \Phi_{\text{smooth}}(\tau_i^a) + \Phi_{\text{int}}(\tau_i^a). \tag{5.1}$$

Details on the three summands are given in Section 4.2.3.

5.2.2 Dataset for DE algorithm

To determine the optimal sets of parameters for SFM, the DE algorithm requires a dataset comprising real human trajectories as input.

In this work, the Thör [2] dataset is used, since is an open-source dataset of human motion trajectories in a controlled indoor environment. Such environment is a laboratory room measuring 8.4 x 18.8 m. As described in Figure 5.3, five goals are strategically positioned within the laboratory room to ensure interactions between pedestrians during the navigation. Moreover, the presence of different obstacles in the laboratory room prevents the possibility of walking directly to the goals in a straight line. To record the scene, a fixed camera is mounted in a corner of the laboratory room.

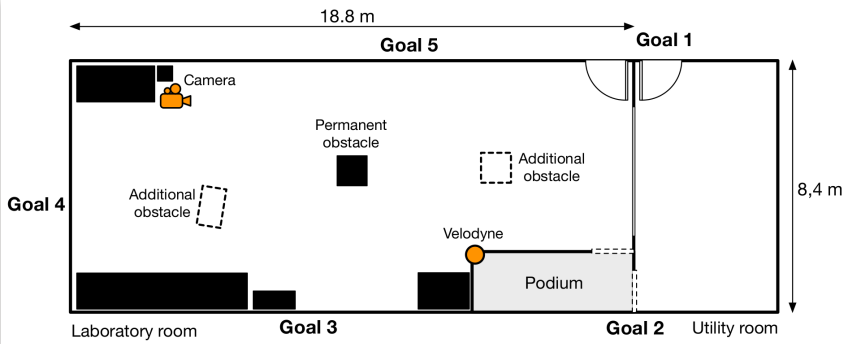


Fig. 5.3 Overview of the laboratory room where the real human trajectories are recorded [2].

To track the motion of humans in real-time, the authors use 10 Qualisys Oqus 7+ infrared cameras (Figure 5.4a) mounted around the perimeter of the laboratory room. Such cameras are used to identify distinct markers on pedestrians' helmets, which can reflect infrared light (Figure 5.4b).

(a) (b)

Fig. 5.4 (a) Qualisys Oqus 7+ infrared cameras; (b) Helmets equipped with reflective markers, designed for pedestrian tracking.

In the Thör dataset three types of scenarios are recorded:

1. *One obstacle:* pedestrians move in the environment without robots, with only a static obstacle positioned at the central of the laboratory room (see Figure 5.5a);

2. *Moving robot:* pedestrians and robot move in the same laboratory room, with a static obstacle strategically positioned at the center of the room. Figure 5.5b illustrates in black the trajectory of the robot, and the remaining colour the trajectory for each participant in the experiment. The robot used in the experiment is programmed to exhibit **socially unaware behavior**, adhering to a predetermined path around the room and maintaining a constant speed and trajectory without adjustments for the presence of surrounding humans;

3. *Three obstacles:* similar to the first scenario, pedestrians move in the laboratory room in the absence of any robots. However, in this case, three static obstacles are present within the room as shown in Figure 5.5c.

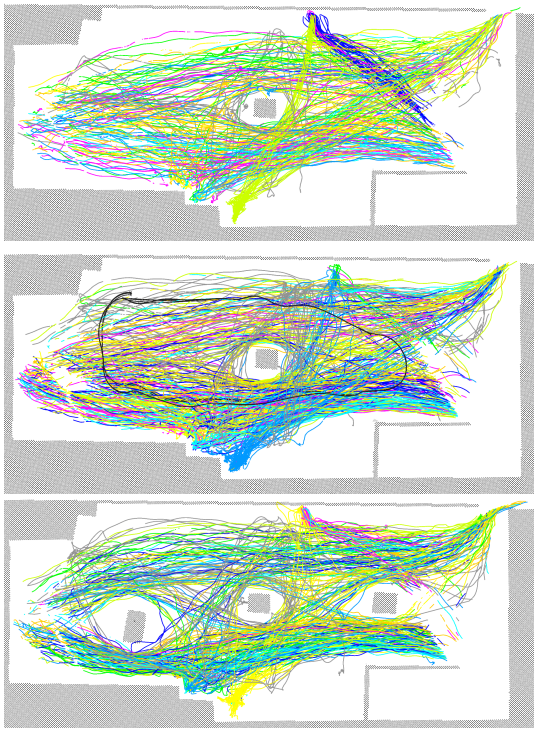


Fig. 5.5 Real trajectories of pedestrians in the three experiments across the three distinct scenarios [2]: (top) *One obstacle* - (centre) *Moving robot* - (bottom) *Three obstacles.*

To compute the best sets of parameters, we exclude the use of the second scenario because the robot moves in an unaware manner. This situation could create some biases in the movement of the agents, resulting in a computation of biased parameters, and we aim to avoid this. Moreover, we also exclude the third case because the recorded trajectories are highly dependent on the placement of obstacles in the room. Thus, to compute the best sets of parameters for the SFM, we used only the first scenario of the Thör dataset, which seems to be the most generalizable scenario among the three.

5.2.3 Differential Evolution Algorithm (DE)

The DE addresses the traditional challenge of optimization problems. Optimization involves the process of discovering the most suitable solution for a problem while adhering to predefined constraints. When optimizing the performance of a system,

the objective is to determine a specific set of parameter values that will lead to optimal system performance under predefined conditions.

Indeed, in this work, the DE is used to estimate the best sets of parameters for SFM that approximate the real human trajectories derived from the Thör dataset.

The DE works with a population comprising *NP D-dimensional real-valued parameter vectors*. *NP* denotes the number of parameter vectors within the population, while D is the number of parameters in each vector. These parameter vectors are called *chromosomes*, and the parameters within them are called *genes*. Each chromosome represents a potential solution to the optimization problem under consideration.

As shown in Figure 5.6, the DE algorithm works through a sequence of phases. In particular, the DE employs common Evolutionary Algorithm (EA) operations such as *mutation*, *crossover*, and *selection* to identify the most promising chromosomes for propagation to the next generation. This iterative process occurs for a specified number of generations, denoted as *NG*.

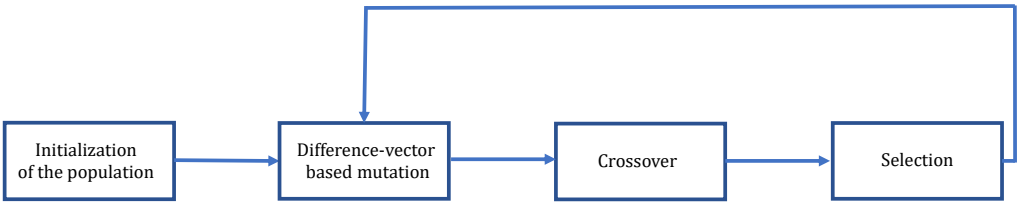


Fig. 5.6 Main phases of the DE algorithm [3].

The *i-th* chromosome in the population, given a generic generation *j*, are represented using the following notation:

$$\boldsymbol{X}_{i,j} = \left\{x_{i,j}^{1},...,x_{i,j}^{D}\right\}, \text{ for } i = 1,...,NP \text{ and } j = 1,...,NG \tag{5.2}$$

where each gene $(x_{i,j})$ is a parameter defined as real number that controls the performance of the SFM.

To select the chromosome that advances to the next generation, an objective function $f(\boldsymbol{X}_{i,j})$ is defined to assess the performance of each chromosome in the

population. This function is custom-designed according to the characteristics of the system under analysis and relies on the parameters within each chromosome.

Thus, the primary objective of the DE is to search for the optimal chromosome, ($\boldsymbol{X}^*_{i,j}$), which guarantees the minimization of the objective function ($f(\boldsymbol{X}_{i,j})$) among all the chromosomes generated in the different generations (NG).

At the state-of-the-art, numerous adaptations of DE exist [127, 128], each distinguished by the specific strategies that the authors employ in the execution of the main phases of the algorithm. In this thesis, the classical DE, developed by Storm and Prince [128], is chosen.

Following the scheme represented in Figure 5.6, the details of each phase are described below.

1. *Initialization of Parameter Vectors:*

   The initial phase of the DE algorithm consists of generating an initial population of parameter vectors. As described above, each chromosome is composed of D real values, which correspond to the physical parameters of the SFM that need to be optimized. These values are subject to constraints defined by the physical nature of the quantity they represent. For example, if a parameter is a mass or a length, negative values are invalid.

   By establishing the boundaries of the intervals for all D parameters, the acceptable value limits is defined as follows:

$$\begin{aligned} \boldsymbol{X}_{min} &= \left\{x^1_{min}, ..., x^D_{min}\right\} \\ \boldsymbol{X}_{max} &= \left\{x^1_{max}, ..., x^D_{max}\right\} \end{aligned} \tag{5.3}$$

   As suggested in [3], the objective of the initial population is to comprehensively span these intervals. This is achieved by uniformly randomizing the chromosomes within the search space, taking into account the predefined boundaries. Therefore, the initial value of the *k-th* parameter, belonging to the chromosome $\boldsymbol{X}_{i,0}$, is determined as follows:

$$x^k_{i,0} = x^k_{min} + rand\,[0,1]^k_i \left(x^k_{max} - x^k_{min}\right), \text{ for } k = 1, ..., D \tag{5.4}$$

where $rand\,[0,1]_i^k$ is a uniformly distributed random number between 0 and 1, generated independently for each parameter (k) of the *i-th* chromosome.

2. *Mutation with Difference Vectors:*

   The mutation is the process whereby every chromosome in the current population, known as *target vector* $\boldsymbol{X}_{i,j}$, undergoes a perturbation to obtain a new mutant chromosome generated from it, known as *donor vector* $\boldsymbol{V}_{i,j}$:

$$\boldsymbol{V}_{i,j} = \left\{v_{i,j}^1, ..., v_{i,j}^D\right\}, \text{ for } i = 1, ..., NP \tag{5.5}$$

   The *donor vector* $\boldsymbol{V}_{i,j}$ is generated by sampling three other chromosomes of the current population, referred to as $\boldsymbol{X}_{r_1^i,j}$, $\boldsymbol{X}_{r_2^i,j}$, and $\boldsymbol{X}_{r_3^i,j}$. These chromosomes are randomly selected by choosing within the range [1, NP]. Moreover, these randomly selected chromosomes are mutually exclusive and distinct from the current chromosome index i.

   During the DE-mutation, the *donor vector* $\boldsymbol{V}_{i,j}$ is computed as follows:

$$\boldsymbol{V}_{i,j} = \boldsymbol{X}_{r_1^i,j} + F\left(\boldsymbol{X}_{r_2^i,j} - \boldsymbol{X}_{r_3^i,j}\right) \tag{5.6}$$

   where the difference between two randomly selected chromosomes is scaled by a factor F, and the scaled difference is added to the first randomly selected chromosome. The mutation procedure is illustrated in Figure 5.7.

   The factor F, called *mutation factor*, has a significant role in the overall performance of the DE algorithm. Indeed, this factor allows adjusting the influence of the difference vector on the mutation of the *target vector*.

3. *Crossover:*

   This phase is indispensable to create diversification among the chromosomes present in the population. In particular, this phase aims to exchange genes contained in a *target tector* $\boldsymbol{X}_{i,j}$ with those associated with the *donor vector* $\boldsymbol{V}_{i,j}$, to generate a new chromosome called *trial vector* $\boldsymbol{U}_{i,j}$:

$$\boldsymbol{U}_{i,j} = \left\{u_{i,j}^1, ..., u_{i,j}^D\right\}, \text{ for } i = 1, ..., NP \tag{5.7}$$

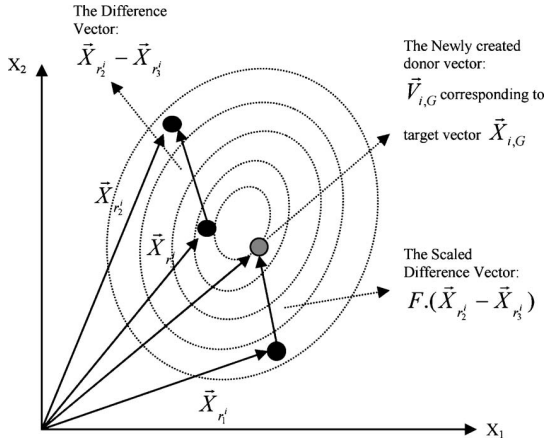


Fig. 5.7 DE-mutation description in a 2-D search space [3].

The number of replaced genes in the *target vector* is determined by another important parameter of the algorithm, namely the *crossover rate Cr*. The strategy adopted for performing crossover is the *binomial crossover*.

The *binomial crossover* is so named because the number of inherited parameters from the *donor vector* exhibit a (nearly) binomial distribution, controlled by the value of *Cr* [3]. Consequently, the *k-th* gene of the *trial vector* $\boldsymbol{U}_{i,j}$ is calculated as follows:

$$u_{i,j}^k = \begin{cases} v_{i,j}^k, \text{ if } (rand_i^k[0,1] \leq CR \text{ or } k = k_{rand}) \\ x_{i,j}^k, \text{otherwise} \end{cases} \tag{5.8}$$

where $rand_i^k$ is a uniformly distributed random number between 0 and 1, which is called for each *k-th* gene of the *i-th* chromosome. Moreover, $k_{rand} \in [1,2,...,D]$ is a randomly chosen index which guarantees that the *trial vector* inherits at least one gene from the *donor vector*. k_{rand} is calculated once for each chromosome per generation *j*. In this way, the *trial vector* $\boldsymbol{U}_{i,j}$ is never the same as the *target vector* $\boldsymbol{X}_{i,j}$.

4. *Selection:*

To keep the population size constant across successive generations, the DE involves a selection process that determines whether the *target* or *trial* vector is maintained for the next generation.

The decision is made according to the objective function value computed for each chromosome in the population:

$$\boldsymbol{X}_{i,j+1} = \begin{cases} \boldsymbol{U}_{i,j}, \text{ if } f(\boldsymbol{U}_{i,j}) \leq f(\boldsymbol{X}_{i,j}) \\ \boldsymbol{X}_{i,j}, \text{otherwise} \end{cases} \tag{5.9}$$

After completing these four phases for each chromosome of the population, a new population of chromosomes is generated. This iterative process continues until the stopping criterion is reached. In our case, the process continues iterative until the maximum number of generations (NG) is reached.

Control Parameters of the algorithm

The DE algorithm performance depends on three key control parameters: the population size (NP), the mutation factor (F), and the crossover rate (Cr). Within this section, the goal is twofold: first, we examine how each of these parameters impacts the overall performance of the DE algorithm and second, we describe how each control parameter is chosen for the implemented DE algorithm.

- ***Population Size NP***: NP is the number of chromosomes in a population. As indicated in [128], usually NP is chosen between 5D and 10 D, where D is the number of genes (parameters) for each chromosome. In our problem, we deal with the estimation of 7 parameters, thus, the population size was chosen equal to 42;

- ***Mutation factor F***: F is the factor used in Equation 5.6 to scale the difference between the two selected random chromosomes. In [128], Storn and Price suggest the range of choice of this parameter between 0.4 and 1. In Figure 5.8 the effects of the F factor on the difference vector are shown. In particular, the F factor has a twofold purpose: first, it prevents to generate a duplication of an existing chromosome (Figure 5.8a); second, it prevents to remain trapped in a local minimum (Figure 5.8b). Indeed, choosing a value of F close to one means increasing the capacity of the *exploration* in the solution space of the algorithm. In our case, the value of the F factor was chosen equal to 0.5;

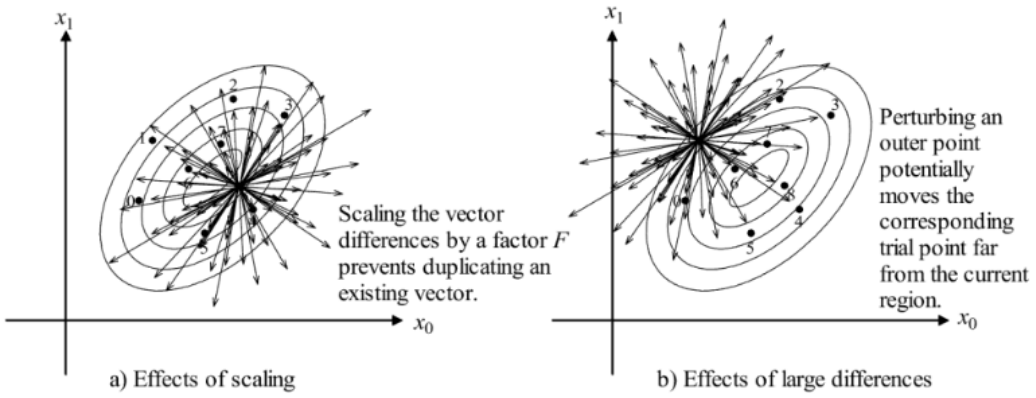


Fig. 5.8 a) Effect of the Factor F in general; b) Effect of increasing the F factor [4].

- ***Crossover rate Cr***: Cr is used in the crossover phase to generate the *trial vector*. In particular, the Cr parameter controls how many genes from the *donor vector* are given to the *trial vector* (see Equation 5.8). As suggested in [3], the Cr is usually between 0 and 1.

  For a low value of Cr (near to 0), a low number of genes from the *donor vector* are transferred to the *trial vector*. Resulting in a minimal alteration of the chromosome parameters from one generation and the other. On the other hand, a high value of Cr (near to 1) leads to the transfer of a high number of genes from the *donor vector* to the *trial vector*. Consequently, the chromosomes in the new generation are more likely to exhibit significant differences when compared to those in the previous generation.

  Thus, in our algorithm, the Cr is chosen equal to 0.6.

In Table 5.1, the values chosen for each of the parameters of the DE algorithm are summarized.

Symbol	Value	Meaning
NP	42	Population size
NG	90	Number of generations
F	0.5	Mutation factor
Cr	0.6	Crossover rate

Table 5.1 Control parameters values used for the implementation of the DE algorithm.

Symbol	Meaning
$A_i \in \mathbb{R}_{>0}$	strength of interaction force for pedestrian i
$B_i \in \mathbb{R}_{>0}$	range of interaction force for pedestrian i
$r_i \in \mathbb{R}_{>0}$	radius of the personal space of pedestrian i
$R_0 \in \mathbb{R}_{>0}$	minimum admissible distance between robot and obstacle
$\lambda \in [0,1]$	strength of anisotropic behavior
$v_i^d \in \mathbb{R}_{>0}$	desired velocity (in modulus) of pedestrian i
$\alpha_i \in \mathbb{R}_{\leq 0}$	relaxation time of pedestrian i

Table 5.2 SFM parameters estimated by the DE.

5.2.4 DE applied on SFM and DE simulation results

Similar to [129], in this thesis the estimation of the SFM parameters has been made exploiting the DE algorithm applied to an open source dataset (Thör [2]).

In particular, Johansson et al. [129] utilize an evolutionary algorithm to estimate SFM parameters using data obtained from video tracking. Similarly, in this work, we used the Thör dataset and we substituted (one at a time) each pedestrian within the experiment with a simulated robot. Such robot is controlled by SFM with the parameters estimated with the DE algorithm. This approach allows a comparison between the positions of the simulated robot and the position of the real pedestrian. The comparison is executed during the *selection phase* of the DE algorithm with the following objective function:

$$f(\boldsymbol{X}_{i,j}) = \frac{\sum_{k=1}^{T_{prev}} \|\mathbf{p}_{robot}(t+k\Delta_t) - \mathbf{p}_{pedReal}(t+k\Delta_t)\|}{N} \tag{5.10}$$

where $\mathbf{p}_{robot}(t+k\Delta_t)$ and $\mathbf{p}_{pedReal}(t+k\Delta_t)$ are the position of the simulated robot and the real pedestrians, respectively. N is the total number of points of the trajectory produced by SFM over a finite time horizon $T_{prev}\Delta_t$.

In Table 5.2 the seven parameters of the SFM estimated with the DE algorithm are summarized.

The main steps to obtain the SFM parameters with DE, applied to the Thör dataset are summarized in Algorithm 6.

The **inputs** of the Algorithm 6 are: the duration of each time-step Δ_t, the number of simulated time-steps T_{prev}, and the Thör dataset. The latter contains different

trajectories for each pedestrian. Every point along each trajectory is identified by a tuple *(x, y, t)*, which contains the coordinates in the 2-D plane and the corresponding time t.

For each real pedestrian (pedReal) belongs to the Thör dataset, the algorithm saves all trajectories in PedTrajectoriesInfo using the function getTrajectoriesInfo (line 3). In this way, all the start, goal positions, and the instant of reaching the goal (T^{tr}_{goal}) for each trajectory (tr) are known. Then, for each trajectory, the DE is applied (line 4) until the time t achieves the T^{tr}_{goal} (line 5). Moreover, at each time t the DE is used to simulate over a fixed time horizon $T_{prev}\Delta_t$. Thus, it is necessary to ensure that the simulated trajectory does not extend beyond the real one (line 6). If the simulated trajectory goes beyond the real one, then the algorithm proceeds to the next trajectory (line 7). Otherwise, the best parameters of the SFM that approximate the real trajectory over the time interval $[t;(t+\Delta_t T_{prev})]$ are estimated (line 8).

Then, the initial position of the robot is set to the position of the real pedestrian (line 9). In line 10, the distance between the robot and the closest obstacle is computed. Then, from line 11 to 13, the distances between the robot and the three nearest pedestrians are computed.

The estimation of the best four parameter sets is accomplished using the function BestEstPar, as indicated in line 15. The inputs of this function are: all the control parameters of the DE algorithm (NP, F, Cr, NG), the boundaries of the intervals for all D parameters ($\boldsymbol{X}_{min}, \boldsymbol{X}_{max}$), and the information regarding the simulation (Δ_t, T_{prev}).

A more comprehensive explanation of this function can be found in Algorithm 7.

Ultimately, the four distances are associated with the four sets of estimated parameters that most accurately approximate that section of the trajectory (line 16). This part is crucial for generating a sufficiently large labelled dataset that can be used as a training dataset for the neural network.

At the end of the algorithm, the time of the simulation has been updated (line 17).

The main goal of the Algorithm 7 is to compute the best sets (bestSets) that approximate the real trajectory through the DE algorithm.

In particular, the DE is iteratively employed four times on the identical segment of the real trajectory (line 2).

Algorithm 6: Main algorithm of SFM parameters estimation with DE in Thör dataset

1 **Input**: Δ_t, T_{prev}, ThorDataset
2 **foreach** pedReal $\in$ ThorDataset **do**
3 PedTrajectoriesInfo $\leftarrow$ GetTragectoriesInfo(pedReal)
4 **foreach** tr $\in$ PedTrajectoriesInfo **do**
5 **while** $t < T^{tr}_{goal}$ **do**
6 **if** $(t + T_{prev}\Delta_t) > T^{tr}_{goal}$ **then**
7 **break**
8 **else**
9 $\mathbf{p}_{robot}(t) = \mathbf{p}_{pedReal}(t)$
10 $d_{robot,obs}(t) = \|\mathbf{p}_{robot}(t) - \mathbf{p}_{obs}\|$
11 $d_{robot,ped_1}(t) = \|\mathbf{p}_{robot}(t) - \mathbf{p}_{ped_1}(t)\|$
12 $d_{robot,ped_2}(t) = \|\mathbf{p}_{robot}(t) - \mathbf{p}_{ped_2}(t)\|$
13 $d_{robot,ped_3}(t) = \|\mathbf{p}_{robot}(t) - \mathbf{p}_{ped_3}(t)\|$
14 $\mathbf{d}_{3closestPed}(t) = [d_{robot,ped_1}(t), d_{robot,ped_2}(t), d_{robot,ped_3}(t)]$
15 bestSets $\leftarrow$ BestEstPar$(NP, F, Cr, NG, \boldsymbol{X}_{min}, \boldsymbol{X}_{max}, \Delta_t, T_{prev})$
16 paramDataset $\leftarrow [d_{robot,obs}(t), \mathbf{d}_{3closestPed}(t)$, bestSets$]$
17 $t \leftarrow t + \Delta_t$

For each set, the algorithm starts by initializing three variables: bestChrom, $\boldsymbol{P}_{curr}$, and bestObjVal (as indicated in lines 3-5).

The bestChrom variable ultimately stores the best chromosome, representing the best parameter set approximating the real trajectory. The $\boldsymbol{P}_{curr}$ is initialized as an empty vector, but during the computation will store the current population. The bestObjVal is used to compare the best chromosome obtained within each generation to the best chromosome across all generations, and it is initially set to a random value.

To generate the initial population, the generateChromosome function is called for each chromosome (lines 6-8). As inputs, the function takes the constraints of each parameter of the SFM ($\boldsymbol{X}_{min}, \boldsymbol{X}_{max}$).

Then, at each generation j, the algorithm initializes an empty vector for the new population $\boldsymbol{P}_{new}$, the donor vector $\boldsymbol{V}_{i,j}$, and the trial vector $\boldsymbol{U}_{i,j}$ (lines 10-12).

Successively, for each chromosome ($\boldsymbol{X}_{i,j}$) belongs to the current population ($\boldsymbol{P}_{curr}$), the DE algorithm performs the Mutation (line 15), the Crossover (line 16),

and the Selection functions (line 17). At the end of this iterative procedure, a new population of chromosomes is generated ($\boldsymbol{P}_{new}$).

The Selection function chooses between the target vector ($\boldsymbol{X}_{i,j}$) and the trial vector ($\boldsymbol{U}_{i,j}$). The chosen chromosome is denoted by $\boldsymbol{X}_{i,j+1}$, and the corresponding value of the objective function is indicated with minObjVal. A more detailed explanation of the Selection function is available in Algorithm 8.

To guarantee the selection of the best chromosome across all generations, after each selection step the corresponding objective function value (minObjVal) is compared to bestObjVal (line 19). If the minObjVal is lower than bestObjVal, the $\boldsymbol{X}_{i,j+1}$ is stored as the best chromosome (bestChrom) and the bestObjVal is updated (lines 20-21).

At the end of this iterative process, the new population is set as the current one $\boldsymbol{P}_{curr}$ (line 22). Afterwards, if j is less than NG the process is iterated from line 10.

When the last generation is achieved, the best chromosome (bestChrom), i.e. the chromosome with the lowest objective function value, is stored in bestSets (line 23).

Algorithm 7: The BestEstPar function implementing the DE algorithm

1 **Inputs**: $F, Cr, NP, \boldsymbol{X}_{min}, \boldsymbol{X}_{max}, \Delta_t, T_{prev}$
2 **for** $n_{set} = 1:4$ **do**
3 bestChrom $\leftarrow zeros(D)$
4 $\boldsymbol{P}_{curr} \leftarrow zeros(D, NP)$
5 bestObjVal $= 1000$
6 **for** $i = 1:NP$ **do**
7 $\boldsymbol{X}_{i,0} \leftarrow$ generateChromosome$(\boldsymbol{X}_{\min}, \boldsymbol{X}_{\max})$
8 $\boldsymbol{P}_{curr} \leftarrow \boldsymbol{X}_{i,0}$
9 **for** $j = 1:NG$ **do**
10 $\boldsymbol{P}_{new} = zeros(D, NP)$
11 $\boldsymbol{V}_{i,j} = zeros(D)$
12 $\boldsymbol{U}_{i,j} = zeros(D)$
13 **foreach** $\boldsymbol{X}_{i,j} \in \boldsymbol{P}_{curr}$ **do**
14 $(\boldsymbol{X}_{r_1^i,j}, \boldsymbol{X}_{r_2^i,j}, \boldsymbol{X}_{r_3^i}) \leftarrow$ SamplingRandomChromosome$(\boldsymbol{P}_{curr})$
15 $\boldsymbol{V}_{i,j} \leftarrow$ Mutation$(\boldsymbol{X}_{r_1^i,j}, \boldsymbol{X}_{r_2^i,j}, \boldsymbol{X}_{r_3^i,j}, F, \boldsymbol{X}_{min}, \boldsymbol{X}_{max})$
16 $\boldsymbol{U}_{i,j} \leftarrow$ Crossover$(\boldsymbol{X}_{i,j}, \boldsymbol{V}_{i,j}, Cr)$
17 $(\boldsymbol{X}_{i,j+1}, \text{minObjVal}) \leftarrow$ Selection$(\boldsymbol{X}_{i,j}, \boldsymbol{U}_{i,j}, \Delta_t, T_{prev})$
18 $\boldsymbol{P}_{new} \leftarrow \boldsymbol{X}_{i,j+1}$
19 **if** minObjVal $\leq$ bestObjVal **then**
20 bestObjVal $=$ minObjVal
21 bestChrom $= \boldsymbol{X}_{i,j+1}$
22 $\boldsymbol{P}_{cur} = \boldsymbol{P}_{new}$
23 bestSets $\leftarrow$ bestChrom

Focusing on Algorithm 8, the Selection function takes as inputs: the target vector $\boldsymbol{X}_{i,j}$, the trial vector $\boldsymbol{U}_{i,j}$, the duration of each time-step Δ_t, and the number of simulated time-steps T_{prev}.

Considering each chromosome (i.e. the target vector $\boldsymbol{X}_{i,j}$, trial vector $\boldsymbol{U}_{i,j}$) one by one (line 2), the setSFMparams function sets the chromosome as the parameters of the SFM to drive the robot (line 3).

For each time step (k), the algorithm computes the distance between the robot and the real pedestrian, and stores it in $\mathbf{d}_{robot,ped}$ (line 5). Afterwards, it advances the robot to the subsequent position by applying the SFM, as outlined in line 6.

Following the generation of the trajectory for a given chromosome (lines 4-6), the algorithm evaluates the objective function value associated with that specific

chromosome using the evaluateObjFunc function, in line 7. The evaluateObjFunc function uses the Equation 5.10 to compute the value of the objective function. This value is saved in the ObjValues vector.

Then, when the objective function values linked to each chromosome are stored in the ObjValues vector, the algorithm chooses the minimum value (minObjVal) and the corresponding chromosome (SelectedChrom) and then returns them to the Algorithm 7, see line 8-10.

Algorithm 8: The Selection function

1 **Input**: $\boldsymbol{X}_{i,j}, \boldsymbol{U}_{i,j}, \Delta_t, T_{prev}$
2 **foreach** *chromosome* $\in [\boldsymbol{X}_{i,j}, \boldsymbol{U}_{i,j}]$ **do**
3 $robot \leftarrow setSFMparams$(chromosome)
4 **for** $k = 1 : T_{prev}$ **do**
5 $\mathbf{d}_{robot,ped} \leftarrow \|\mathbf{p}_{robot}(t + k\Delta_t) - \mathbf{p}_{pedReal}(t + k\Delta_t)\|$
6 $\mathbf{p}_{robot}(t + (k+1)\Delta_t) \leftarrow$ ComputationOfNextPositionWithSFM()
7 ObjValues $\leftarrow$ evaluateObjFunc($\mathbf{d}_{robot,ped}$)
8 (minObjVal, SelectedChrom) $\leftarrow$ min(ObjValues)
9 $\boldsymbol{X}_{i,j+1} \leftarrow$ SelectedChrom
10 **return** $\boldsymbol{X}_{i,j+1}$, minObjVal

In Figure 5.9, some results of the DE algorithm applied to the Thör dataset are shown. In particular, in green is shown the real human trajectory and in black the trajectory generated by the SFM with the parameter estimated by the DE algorithm.

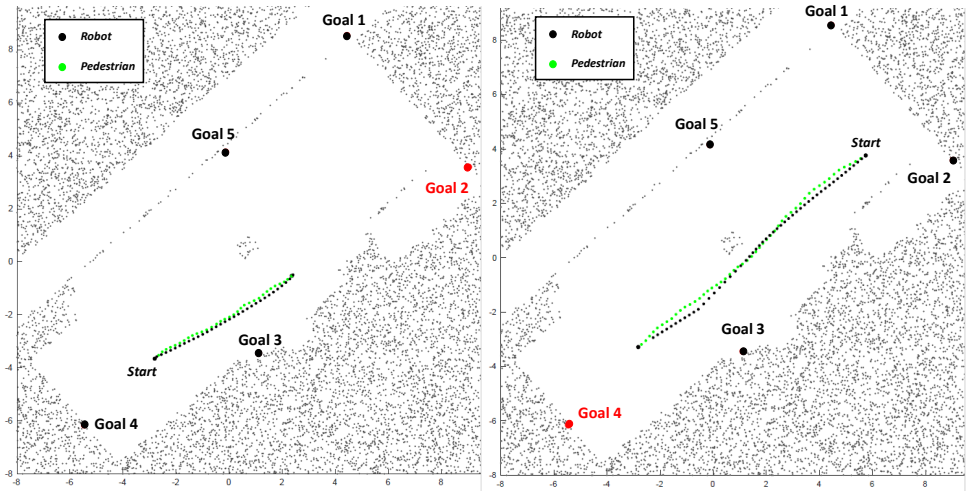


Fig. 5.9 Comparison between the real human trajectory (in green) and the trajectory generated by the SFM with the parameters obtained through the DE (in black).

5.2.5 Real-time parameters estimation through Neural Network

The training and testing phase of the Neural Network (NN), and the use of the NN in the GTSFM algorithm are summarized in Figure 5.10.

In particular, for the creation of the NN, a training labelled dataset obtained from the DE algorithm is used. Then, a testing phase is performed, using another dataset obtained from the DE algorithm (Figure 5.10a). Once the model has been trained and tested, it is ready to be used in the GTSFM algorithm (Figure 5.10b). Indeed, the GTSFM algorithm uses the NN to receive different sets of parameters to generate, through the SFM, the set of trajectories for each agent. Then, the best trajectory for each agent is chosen through the game theory logic.

In this study, the NN has been trained and tested using TensorFlow [130], an open-source machine learning framework developed by Google Brain. Then, the final NN was integrated into ROS in order to ensure the integration into the navigation algorithm of the mobile robot.

In Figure 5.10b, the *inputs*, the *outputs* and the *inner structure* of the NN are highlighted. Starting from the description of the *inputs*, the NN has four input parameters: the distance between the considered agent i and the nearest obstacle, and the distances between the same agent i and the three nearest pedestrians. As a result of this setup, the input layer comprises four neurons.

The *outputs* are a total of 28 parameters, namely 7 parameters for each set, which in this case are 4. As a result of this setup, the output layer comprises 28 neurons. In Figure 5.10b, for the purpose of illustration, 4 neurons are represented as outputs of the NN, indicating the 4 sets.

Regarding the *inner structure*, the NN includes three hidden layers containing 200 neurons each. These neurons employ the ReLU (Rectified Linear Unit) function as activation function, to elaborate the combination of inputs from the preceding layer.

To obtain the final NN, the training and testing phases must be executed. In both phases, the same metric has been used to evaluate performances, i.e. Mean Absolute Error (MAE). In this thesis, the training phase was executed leveraging the *adam* optimizer, an optimization algorithm that employs a stochastic gradient descent method. In this phase, a dataset of 420 labelled data is used. In total, the NN

is trained for 100 epochs, obtaining a final MAE of 0.2665. In the testing phase, a testing dataset of 100 labelled data is used, obtaining a final MAE of 0.2676.

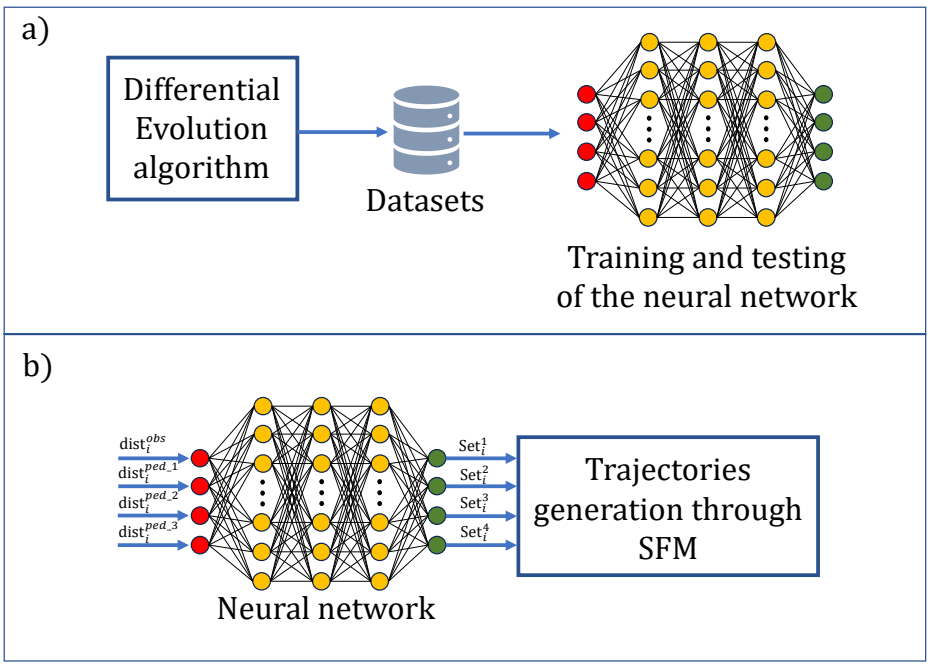


Fig. 5.10 a) Training and testing of the neural network; b) Neural network in the GTSFM algorithm.

5.2.6 Integration of GTSFM algorithm within ROS

This section outlines the implementation details of integrating the GTSFM algorithm into Robotic Operating System (ROS).

For mobile robot navigation, ROS provides the navigation stack framework, where the main component is the *move_base* package. The primary role of the *move_base* is to take a goal pose as input and calculate the necessary linear and angular velocity commands to attain the specified final pose [131]. As shown in Figure 5.11, the *move_base* package is composed of different nodes. Since our goal is to create a motion planning that can generate socially acceptable trajectories, we focus mainly on the local planner node. Thus, to implement the GTSFM without the NN in ROS, we substitute the local planner node with our GTSFM algorithm through the design of a new plugin.

Instead, the design and the integration of the NN with GTSFM within ROS can be summarized in the following three steps:

1. ***Building the Neural Network:***

   for the development, training and testing of the NN, the TensorFlow framework is used. At this stage, the NN is completely independent from ROS.

   In TensorFlow is essential to set different characteristics of the model: number of hidden layers, number of neurons for each layer, activation functions, cost function and optimization method for the training phase.

   After testing the model, it can be saved by storing the weights and biases of its different layers obtained during the training phase. This enables the reloading of the same model, even across different code scripts.

2. ***Creating a ROS node:***

   to integrate the NN into the ROS environment, a straightforward standard node needs to be established. Upon initialization, the node imports the NN saved in the previous step. This enables the model to leverage standard ROS functionalities and interfaces, facilitating communication with other components of the robot.

3. ***Design of ROS interface:***

   to enable the communication between the local planner (GTSFM) and the NN a service communication is implemented. In this way, the NN acts as a server, performing the relevant prediction, and providing as output the parameter sets required for the GTSFM algorithm.

In Figure 5.11, the system architecture and the communication interface between the NN and the move base package are shown.

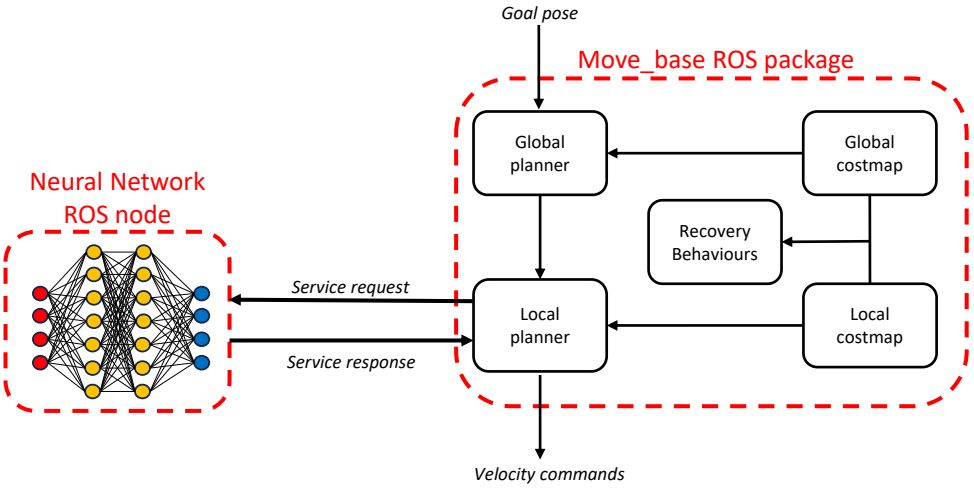


Fig. 5.11 Representation of the system architecture composed by the *move_base* ROS package and the neural network ROS node.

5.3 Simulation setup

Before conducting the real-world experiment, it is crucial to evaluate through *quantitatively* measurement the developed navigation algorithm. To achieve this objective, a series of numerical simulations are conducted to rigorously assess its performance. The simulated scenario mimics the real-world experiment, where different pedestrians and a robot share the same space during navigation. In particular, the simulations are executed on a simulated mobile robot, ideally identical to the one used in the real-world experiment. Leveraging the Locobot WX250s, available in our Complex Systems Laboratory, we use its simulated model within Gazebo to evaluate its navigation capabilities.

The following sub-sections outline the methods employed in the simulation campaign. Initially, an a-priori power analysis is discussed, which estimates the minimum number of simulated trajectories required to obtain statistically significant results. Subsequently, the two simulation software platforms (Gazebo and Rviz) utilized in the experimental simulations, are introduced. Furthermore, a comprehensive description of the simulation scenario is provided, ensuring the interaction between robots and pedestrians. Finally, the last section of this chapter presents the *quantitatively* results of the simulation campaign.

5.3.1 A-priori power analysis for simulation campaign

The final goal of the simulation campaign is to reject the following null hypothesis:

Null Hypotesis All experimental conditions (GTSFM, SFM, ORCA) are statistically indistinguishable considering the state-of-the-art performance metrics testing social navigation.

To estimate the minimum number of trajectories and thus obtain an acceptable statistical result to reject the null hypothesis, we performed an a-priori power analysis [95] using the free software "G*Power" [96]. First, we identified our analysis case as a nonparametric study, since as stated in [97] nonparametric statistics do not place constraints on the distribution of the data. Then, we assumed that the data collected after the a-priori study would be analyzed via the Kruskal-Wallis test. This choice was motivated by the non-parametric nature of the test and our intention to compare 3 independent groups (GTSFM, SFM, ORCA).

Based on [95], we computed the total sample size considering the ANOVA test [98], i.e., the parametric-equivalent test of the Kruskal–Wallis one and then multiplied the result by the corrective factor ARE, obtaining the equivalent sample size of the non-parametric Kruskal–Wallis test. The result of the a-priori analysis for our non-parametric test is about 152 trajectories for each group, considering an alpha level equal to 5%, power of the study 80% and the three groups, corresponding to the three different experimental conditions.

5.3.2 Simulation tools

Gazebo

To simulate the navigation of agents and the scenario, where various pedestrians and a robot coexist within the same environment, we decided to use Gazebo.

Gazebo [132] is an open-source simulator that is commonly used in robotics in conjunction with ROS. Gazebo enables the modeling of the physical behavior of various types of robots, including mobile robots, drones, and anthropomorphic robotic arms. Moreover, Gazebo allows the integration of virtual set of sensors such as cameras, lidar, and ultrasound sensors. These sensors deliver realistic data mimicking real-world sensors. Gazebo, coupled with ROS, facilitates a realistic

simulation environment for robots by enabling control through standard ROS nodes and communication protocols (topics and services). By leveraging Gazebo's feature, researchers can design, test, and refine robotic system functionalities within a virtual environment before real-world deployment. This simulation-based approach offers significant advantages, including accelerated development timelines and reduced costs associated with physical prototyping and real-world experimentation.

Considering all the reasons mentioned above, Gazebo has emerged as the ideal simulator for our simulation campaign to test the developed algorithm.

SFM plugin for pedestrians in Gazebo

To simulate a real-world scenario, which involves multiple pedestrians and a robot navigating in the same environment, one of the essential requirements is to identify a tool capable of simulating pedestrians and their movements.

In its default configuration, Gazebo provides the option to incorporate pedestrians, called "actors", into the simulation. Within the virtual environment, simulated actors initiate from designated starting points and navigate towards designated endpoints along predefined trajectories of varying complexity. However, this conventional approach does not guarantee that the simulated pedestrians exhibit "rational" behavior. Indeed, the simulated pedestrians adhere to their predefined trajectories, even in the presence of potential obstacles, such as walls or other pedestrians. This results in a form of navigation that significantly deviates from the behavior of a real pedestrian.

In [82], the authors introduce a framework for simulating scenarios involving both robot and pedestrians, with a focus on evaluating social navigation. This framework uses real-world pedestrian trajectories, coming from UCI [92] and ETH datasets [1], to simulate pedestrians' navigation. However, a key limitation of this simulator lies in its inability to model the dynamic influence of robot motion on pedestrian trajectories. Furthermore, this simulator lacks integration with the ROS, and the robot used in the scenario cannot be customized.

Although the presented framework offers valuable functionality, it is unable to satisfy our specific requirements for the simulation of realistic human-robot interactions. In particular, our research requires pedestrians with responsive capabilities also with the robot, to accurately model the interactions. Furthermore, to ensure a real simulation of the real-world experiment, we need a comprehensive simulation

that includes both the physical characteristics of the robot and the communication, tools and functionalities provided by ROS. Unfortunately, the limitations of this framework prevent it from aligning with our main research goals.

Therefore, a different methodology is employed to simulate pedestrians and ensure realistic interactions between agents during the navigation. In particular, an alternative solution is to use a plugin in Gazebo. A plugin is a software that permits users to incorporate specific functionalities into the simulator. In particular, in our simulation campaign an open-source SFM plugin [133] is implemented to replicate pedestrian movement within Gazebo. The SFM plugin is based on the original model established by Helbling et al. [20] and its subsequent extensions [134] [135].

This plugin allows pedestrians to respond actively to the robot's presence, mimicking real-world interactions. This responsiveness is achieved by modeling pedestrian movement according to the principles of the SFM.

Moreover, within the simulation campaign, each agent possesses its set of individual parameters. This approach ensures a more realistic and customizable simulation than the default configuration.

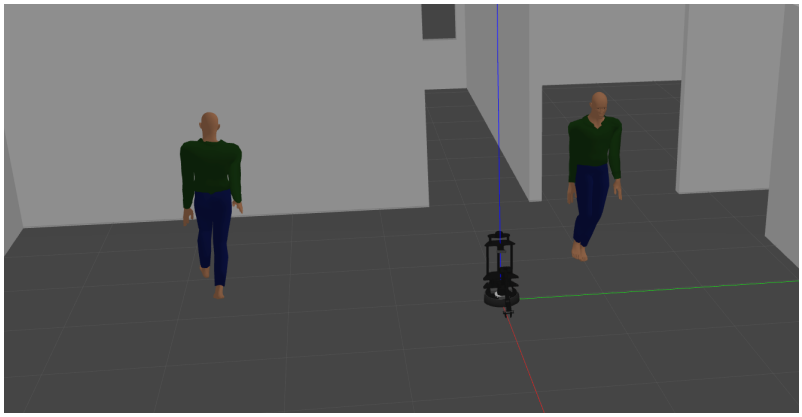


Fig. 5.12 A scenario with a robot and two pedestrians simulated using the SFM plugin within the Gazebo environment.

RViz (Robot Visualization) tool

In our simulation campaign, we used also a 3D visualization tool included in ROS, known as RViz [136]. An example of RViz in our simulation is shown in Figure 5.13.

We decided to use RViz because it provides different functionalities such as:

- ***Real-time visualization:*** RViz allows the graphical visualization of a diverse set of data coming from sensors (e.g. LiDAR and cameras). This data is exchanged within the topics between different active nodes. Furthermore, RViz enables users to design and visualize the model of a robot and any other objects that might be present within the environment;
- ***Planning and simulation:*** This tool offers the capability to dynamically specify the goal pose that the robot needs to achieve and to display it directly on the map. By accessing particular topics, RViz allows the visualization of the path generated by the global planner, the adjustments made by the local planner during navigation, and the real-time movement of the robot within the environment;
- ***Debugging:*** Since RViz allows the data visualization and monitoring of the real-time behaviour of the robot, it serves as a versatile tool for debugging and testing, applicable in simulated environments (e.g. Gazebo) and real robots.

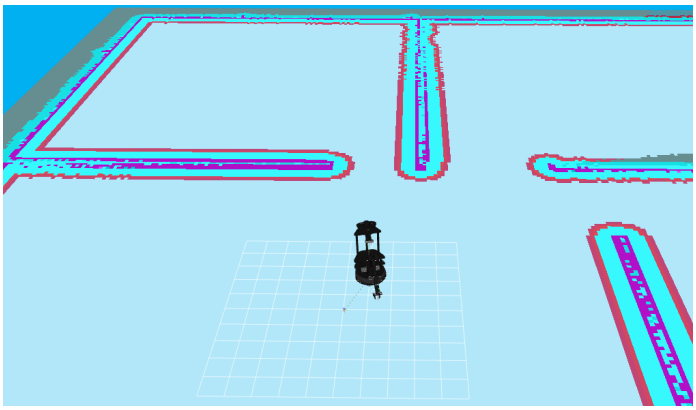


Fig. 5.13 Example of the visualization of the Locobot and the walls through the global costmap in RViz.

5.3.3 Simulations description

The primary objective of the simulation campaign is to evaluate the effectiveness of the designed navigation algorithm considering the state-of-the-art performance metrics testing social navigation (for details see Section 2.3). In particular, we use the PLR and AS to evaluate the overall efficiency of the algorithm. Then, we use

the CPD and the PR to measure the comfort and the smoothness of the trajectories, respectively.

To evaluate the GTSFM motion planner, a comparative analysis is performed, involving two state-of-the-art algorithms: SFM [20] and Optimal Reciprocal Collision Avoidance (ORCA) [21].

The simulation campaign includes 180 simulations (trials) for each of the algorithms under consideration. The rationale behind selecting 180 trials is presented in Section 5.3.1, where the a-priori power analysis is described to estimate the minimum number of trajectories to obtain an acceptable statistical result to reject the null hypothesis. Moreover, in Section 5.3.1, it is stated that the estimated minimum number of trajectories required for each algorithm is 152. However, a slightly larger number, specifically 180, is chosen for our purposes.

The 180 trials consist of two distinct navigation scenarios: the first scenario, consisting of 90 trials, involves three pedestrians, while the second scenario, also comprising 90 trials, incorporates four pedestrians. Both scenarios have been designed to replicate navigation situations characterized by low crowd density conditions, while simultaneously introducing a degree of variability in potential interactions between robots and pedestrians. In these scenarios, all pedestrians navigate from an initial position to a final destination using the SFM Gazebo's plugin presented in Section 5.3.2. The deployment of this plugin is crucial for enabling simulated pedestrians to exhibit avoidance behaviors when encountering the robot, thereby enhancing the authenticity of assessments related to socially-aware navigation.

The simulation environment is created in Gazebo and characterized by the following dimensions: 8.5m x 5.5m. Thus, the overall area comprises approximately 47 m^2, which is consistent with the dimensions of environments commonly referenced in literature for conducting real-world experiments [74] [137]. Furthermore, this size closely aligns with the dimensions of the real room employed for testing the algorithms in a real-world context.

To guarantee that each simulation involves a minimum of one interaction between pedestrians and robot the following methodology is adopted. *First*, the simulation environment is partitioned into six distinct fictitious zones, as shown in Figure 5.14.

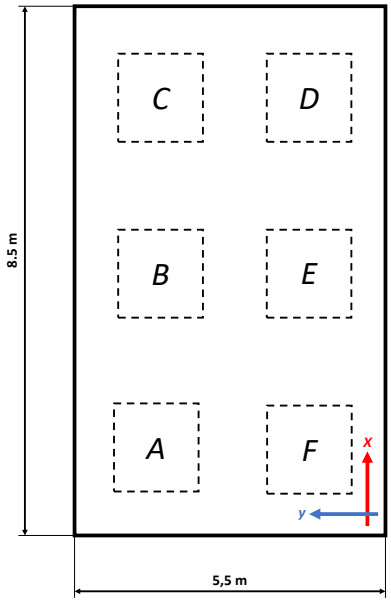


Fig. 5.14 Simulated environment with six fictitious zones.

Specifically, each fictitious zone z is defined by a set of coordinates ranges:

$$\begin{aligned} \mathbf{Z}_x &= \{x^z_{min}; x^z_{max}\} \\ \mathbf{Z}_y &= \{y^z_{min}; y^z_{max}\} \end{aligned} \tag{5.11}$$

Second, the spawn zones, denoting the simulation starting points, and the goal zones are strategically configured for each agent to guarantee the occurrence of interactions between agents, as summarized in Table 5.3.

Spawn zone	Goal zone
A	D, E
B	D, F
C	E, F
D	A, B
F	C

Table 5.3 Each spawn zone is paired with corresponding goal zones.

For each trial, zone F is utilized to spawn the robot during its initialization within the simulated environment. On the other hand, zones A, B, C, D are designated for the spawning of pedestrians.

Before the start of each trial, each pedestrian i, within the simulated environment, autonomously selects its spawn zone z from the available options, using a uniform distribution. Then, each pedestrian i stochastically determines its initial position within the chosen zone z, with the following equations:

$$x_i^z = x_{min}^z + rand(0,10)\frac{x_{max}^z - x_{min}^z}{10} \tag{5.12}$$

$$y_i^z = y_{min}^z + rand(0,10)\frac{y_{max}^z - y_{min}^z}{10} \tag{5.13}$$

where $rand(0,10)$ is a uniformly distributed random number ranging from 0 to 10. Instead, $x_{min}^z, x_{max}^z, y_{min}^z, y_{max}^z$ are the ranges of the chosen zone z.

The spawn zones are *mutually exclusive*, indicating that once a zone is chosen as a spawn location by one pedestrian, it becomes unavailable for selection by another pedestrian.

After determining its initial position, each pedestrian proceeds to randomly select the goal zone based on the principle outlined in Table 5.3. Subsequently, each pedestrian performs the calculation to determine the final goal position to reach within the chosen zone. This process is carried out using the same method employed for the calculation of initial position described in Equations 5.12 and 5.13. In this context as well, the chosen goal zones are *mutually exclusive* to prevent the occurrence of multiple pedestrians sharing a common target area.

5.4 Simulation results of the GTSFM trajectory planning

The outcomes of the performance metrics, obtained through a Monte Carlo numerical simulation, are shown in Figure 5.15.

For each performance metric, a Kruskal-Wallis test is conducted to statistically reject the following null hypothesis:

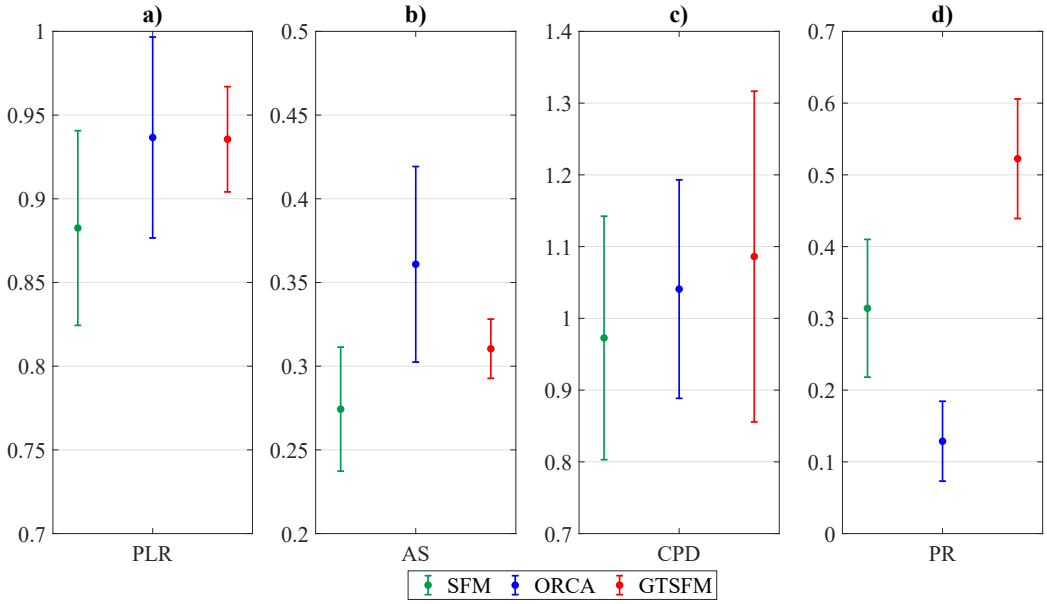


Fig. 5.15 Mean value and standard deviation of the considered performance metrics for each algorithm: SFM (Social Force Model), ORCA (Optimal Reciprocal Collision Avoidance), GTSFM (Game-theoretic Social Force Model). The performance metrics are: a) PLR (Path Length Ratio), b) AS (Average Speed), c) CPD (Closest Pedestrian Distance), d) PR (Path Regularity).

Null Hypotesis All algorithms (GTSFM, SFM, ORCA) are statistically indistinguishable considering the state-of-the-art performance metrics testing social navigation.

Then, since for each performance metric, a statistically significant difference is identified ($p < 0.05$, $\chi^2 = 173,22$ for PLR; $p < 0.05$, $\chi^2 = 222,00$ for AS; $p < 0.05$, $\chi^2 = 53,18$ for CPD; $p < 0.05$, $\chi^2 = 435,16$ for PR), a post-hoc analysis, involving pairwise comparison of algorithms, is conducted and below is discussed.

In general, our numerical validation confirms that GTSFM exhibits a significant improvement over SFM across all four performance metrics.

In particular, in Figure 5.15a, the PLR for each algorithm is shown and reveals that the SFM is characterized by the lowest average PLR, followed by ORCA and GTSFM in a tie. Although ORCA has a very similar mean value to GTSFM, post-hoc analysis reveals a presence of significant statistical difference ($p = 0.02$). This result is justified by the fact that the post-hoc analysis compares the mean of the

calculated ranks and not the mean of the values (mean rank SFM= 148,11; mean rank ORCA=353.59; mean rank GTSFM=309.79). When examining the remaining pairs of groups, a statistically significant difference is observed (SFM-ORCA $p < 0.05$; SFM-GTSFM $p < 0.05$). Regarding the standard deviation, ORCA and SFM exhibit a higher variability compared to GTSFM. This result indicates that in terms of PLR, the repeatability and predictability of each experimental condition are much higher in GTSFM compared to ORCA and SFM.

The average speed (AS) in the three experimental conditions is illustrated in Figure 5.15b, where the highest average value is related to ORCA, followed by GTSFM and SFM, which are in turn distinguishable from one another (post-hoc analysis: for SFM-ORCA p<0.05; for SFM-GTSFM $p < 0.05$; for ORCA-GTSFM $p < 0.05$). The highest mean value of ORCA is justified by the fact that ORCA is designed to maximize the velocity of the robot maintaining a safe distance from other agents. GTSFM maintains a higher mean value than SFM because one of the distinctive features of GTSFM lies in its predictive logic. Unlike SFM, which tends to avoid agents when it is nearby, GTSFM adopts a strategy of reducing speed and selecting a trajectory that avoids people in advance. Similar to SFM, ORCA is a reactive method and thus fails to predict the trajectories of other agents along a consistent time horizon. This limitation results in higher standard deviations for ORCA and SFM compared to GTSFM, suggesting their potential for less stable navigation in dynamic environments. This result highlights the fact that the average speed of the GTSFM remains about the same in the 180 trial indicating a greater repeatability capacity of the GTSFM than the other two algorithms.

Although the results are promising, the average speed values obtained are not yet comparable to human speed values. This limitation is tied to the physical constraints of the Locobot, which has a maximum speed limit of 0.5 m/s.

Average values of CPD in the three experimental conditions are illustrated in Figure 5.15c, where the highest average value is related to GTSFM, followed by ORCA and SFM. The reason for this ranking is presumably due to the purely reactive design of the ORCA and SFM algorithm. Although GTSFM has a higher mean value than ORCA, post-hoc analysis reveals an absence of significant statistical difference ($p = 0.79$), proving that both algorithms try to maintain a certain distance from people to ensure their comfort. When examining the remaining pairs of groups, a

statistically significant difference is observed (SFM-ORCA $p < 0.05$; SFM-GTSFM $p < 0.05$).

The metric that most clearly underlines the significant advantage of employing game theory in navigation is path regularity (PR). In Figure 5.15d, the average PR for each experimental condition is illustrated, where the highest average value pertains to GTSFM, followed by SFM and ORCA, each of which is distinguishable from the others (post-hoc analysis: for SFM-ORCA $p < 0.05$; for SFM-GTSFM $p < 0.05$; for ORCA-GTSFM $p < 0.05$). The highest mean value of the GTSFM is probably justified by the incorporation of game theory enables the identification of optimal parameters within the SFM to execute evasive maneuvers as smoothly as possible.

Chapter 6

Qualitative evaluation of the GTSFM algorithm through a real-world experiment

In this chapter, the focus is detailing the real-world experiment designed to *qualitatively* assess the GTSFM algorithm presented in Chapter 5. This chapter delves into various aspects of the experiment, such as the hardware, the questionnaires to gather subjective feedback, and the statistical analysis to provide significant statistical results. Then, the chapter proceeds to present the result of the experiment and engage in a thorough discussion of the *quantitative* and *qualitative* findings, aiming to provide insights into the algorithm's performance and effectiveness across the different analyses.

6.1 Hardware description

To validate *qualitatively* the trajectories generated by our motion planning we conduct a real-world experiment. This experiment entails the utilization of the Locobot WX250s (Figure 6.1). Additionally, an external camera (Zed2) is employed to monitor participants and acquire data on their positions and velocities. Therefore, the primary objective of this section is to provide a comprehensive description of the hardware components of the Locobot WX250s and the camera employed in the context of the real-world experiment.

Fig. 6.1 The mobile robot (Locobot WX250s-6DOF) used in the real-world experiment.

6.1.1 Mobile robot hardware

The real-world experiment employs the Locobot WX250s (Figure 6.1), developed and commercialized by Trossen Robotics. The system is composed of several components, each of which is essential for its functioning. In the following, a detailed description of each component is provided.

Mobile base

The mobile base is the hardware that facilitates the robot's mobility within its environment. The specific platform utilized as the mobile base is the Create3, developed by iRobot. This platform is designed to be compatible with various connectivity options such as Wi-Fi, Bluetooth, Ethernet, or USB connections [5].

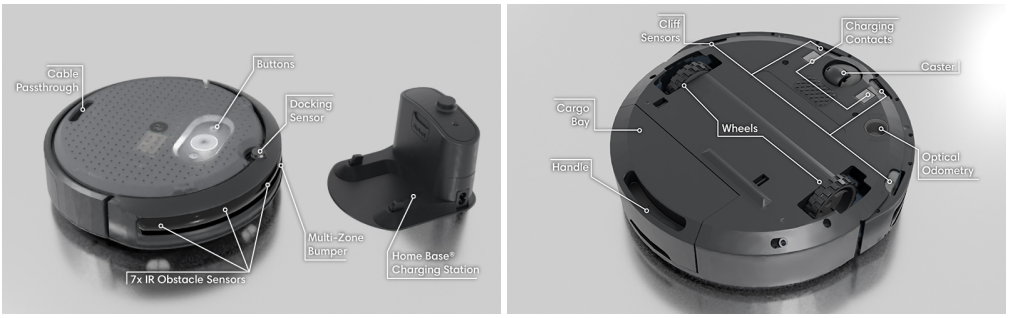


Fig. 6.2 Create3 mobile base [5] from above (on the left), from the bottom (on the right).

The entire software of the mobile base is implemented using ROS2, which means that sensor data and actuator commands are transmitted or received through this framework. To ensure compatibility with ROS1, the mobile base uses a "bridge" that facilitates real-time communication among the different versions of ROS. This strategy allows the development and deployment of navigation algorithms in ROS1 into the mobile base [138]. Thanks to this bridge, all the algorithms were developed using ROS1, and then, we effectively conducted the real-world experiment on our robot, achieving successful results.

The mobile base has two independent drive wheels, as shown in Figure 6.2 on the right. Moreover, to enhance stability during the motion, the mobile base has a "caster", strategically positioned to shift the centre of gravity to its vertical axis, because by default the centre of gravity is shifted forward to its vertical axis. Furthermore, each wheel has current sensors, encoders, and odometry sensor. Data collected from these sensors is integrated with information from the Inertia Measurement Unit (IMU) to produce a unified odometry estimate [139].

Intel NUC Mini PC

The processing system of the robot, responsible for enabling its operations and supervising the sensors, is a NUC computer equipped with the following specifications: 8th Gen Intel Dual-Core i3, 8GB DDR4 RAM, 240 GB Solid State Drive (SSD), Intel Iris Plus Graphics 655, Wi-fi, Bluetooth 5.0, Gigabit Ethernet, USB, Thunderbolt 3, Ubuntu 20.04 [140].

Fig. 6.3 Intel NUC Mini PC.

Unlike the mobile base, the NUC is powered by an independent 50000 mAh secondary battery, which is responsible for powering other robot components, as described below.

WidowX-250 Robot arm

WidowX-250 Robot arm (Figure 6.4) is a manipulator with six degrees of freedom and it is controlled using nine smart servos from DYNAMIXEL-X Series Actuators developed by Robotis. These features provide excellent manoeuvrability, enabling a maximum reach of 650 mm and a complete 360° rotation [141].

Fig. 6.4 WidowX-250 6DOF Robot arm.

Since our primary focus is the investigation of social navigation without the need for object manipulation, the robotic arm was not employed for the scope of the real-world experiment.

RPLIDAR A2M8

The RPLIDAR A2M8 is a 360-degree 2D LiDAR system designed specifically for indoor environments (Figure 6.5). It is positioned on the top of the robot and with its

rapid rotation, it can capture up to 8000 laser-ranging samples per second within a range of 12 meters [138].

The resulting 2D point cloud data produced by the Lidar is used for dual purposes: (i) for Simultaneous Localization and Mapping (SLAM); and (ii) for navigation. In the context of our real-world experiment, this LiDAR system is used for both purposes. Indeed, it facilitated the precise execution of the SLAM process and the detection of the nearest obstacles.

Fig. 6.5 RPLIDAR A2M8.

The main limitation of the LiDAR is its incapacity to offer vertical information. Since the LiDAR is a 2D sensor, it can scan only in the plane perpendicular to its rotational axis. Consequently, it cannot detect obstacles lower or higher than its scanning plane, which restricts its capacity to fully explore and recognize the environment. For this reason, an additional sensor has been placed on the robot that allows for information even below and above the LiDAR scanning plane: Intel RealSense Depth Camera.

Intel RealSense Depth Camera D435

The RealSense Depth Camera D435, an Intel product, is a 3D camera designed to deliver real-time depth data and reconstruct three-dimensional environments (Figure 6.6). It achieves this by utilizing two synchronized stereoscopic cameras to capture images from different perspectives. Moreover, the RealSense has a depth sensor with infrared that measures the distance between the camera and the objects within the scene. This capability allows for the acquisition of precise depth maps of the captured objects. This camera is primarily employed in robotics, virtual reality,

and augmented reality applications. Notably, it demonstrates versatility by being suitable for use in both indoor and outdoor settings [142].

Fig. 6.6 Intel RealSense Depth Camera D435

The RealSense is placed on the Locobot, approximately 53 cm above the floor, and its primary application is during the SLAM phase. Following this phase, the data collected from the camera is merged with the data from the LiDAR to generate a static map, which will be utilized for real-time navigation.

6.1.2 Zed2 camera

Since a Social Navigation algorithm requires real-time data of pedestrians' positions and velocities within the experimental environment, pedestrian tracking becomes indispensable.

Although the robot is equipped with a stereo camera that provides visual information about its surroundings, it cannot track all pedestrians participating in the experiment. Furthermore, the provided information could be partial and limited by the camera's field of view. Consequently, pedestrians positioned outside this field of view remain unidentified by the camera.

To overcome this problem and thus protect the safety of our participants, we use a second camera, the ZED2 Camera (Figure 6.7). The ZED2 camera is designed by Stereolabs to produce a stereoscopic 3D image, mimicking the way human vision works.

Fig. 6.7 Zed2 camera used in real-world experiment.

The main characteristics of the ZED2 are:

- ***Stereo camera:*** as shown in Figure 6.7, the ZED2 is equipped with two synchronized cameras that capture images from different prospective, closely emulating the functioning of the human eye. This stereoscopic configuration allows the camera to estimate *depth* and *motion* information analyzing the pixel displacement between the left and right images [143];
- ***Resolution and frame rate:*** both cameras can be calibrated to different resolution settings. Depending on the resolution set, there will also be a maximum frame rate limit [144];
- ***Field of view:*** The camera has a wide field of view of approximately 110°(H) x 70°(V) [143];
- ***Modes of use:*** The ZED2 can be used to perform various operations such as:
  1. *Positional tracking (or Motion tracking):* this mode enables the camera to estimate its position in the world. This tracking can be used in real-time to track the device on which the camera is mounted [145];
  2. *Spatial mapping:* this mode allows the generation of a 3D map of the environment [146];
  3. *Object detection:* this mode enables the camera to identify objects in the image. The stereo camera can identify only limited object classes, such as *people*, *vehicle*, *bag*, *animal*, *electronics*, *fruit and vegetables*. Moreover, using this mode, the camera can provide the 2D and 3D coordinates of the identified object [147];
  4. *Body tracking:* the identification of some specific keypoints of human body helps in reconstructing a representation of the people's skeleton in

the image. Then, by utilizing the depth and positional tracking performed by the previous mode of the camera, the 3D position of each keypoint can be determined [6]. An example of this mode is shown in Figure 6.8.

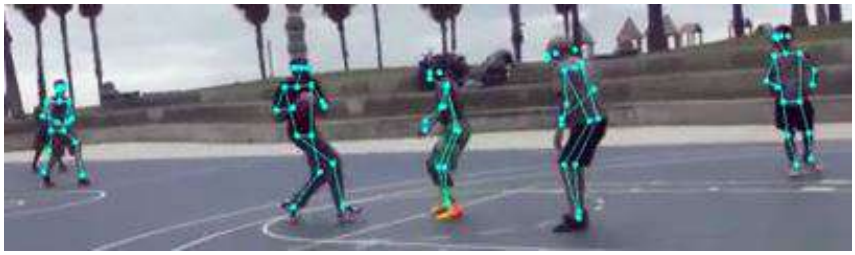


Fig. 6.8 An example of the body tracking mode of Zed2 [6].

The camera offers compatibility with development in both ROS1 and ROS2 [148]. This flexibility allows the the stereo camera to communicate with the Locobot framework, as both are ROS compatible.

People detection and tracking with Zed2

During the real-world experiment, the *object detection* mode (specifying the class *people*) is used to identify pedestrians in the environment. Then, the corresponding people coordinates, referred to the static map generated in the SLAM process, are obtained.

Below, a detailed description of people detection mode is provided [147].

1. ***Data acquisition:*** the Zed2 captures images from different angles using its two cameras. These images are exploited to obtain depth data, which allows the camera to reconstruct a three-dimensional map of the scene;
2. ***People recognition:*** the Zed2 attempts to identify the objects present in both left and right cameras using AI and neural networks and computes their 3D position and velocity;
3. ***Tracking initialization:*** In this phase, the system initiates the tracking process, which involves assigning a unique identifier to the detected person and estimating a bounding box. The bounding box is initially obtained in 2D by recognizing individual pixels that belong to the boundary of the identified

person. Subsequently, the equivalent 3D bounding box is derived from the integration of the 2D version with depth information;

4. ***Object tracking:*** Using depth data and observing its variation over time, the Zed2 camera is able to estimate the distance, direction and speed of a detected object's movement in three-dimensional space, in real time.

6.2 Questionnaire

To evaluate how humans perceive robots controlled by the GTSFM algorithm, the Human-Robot Interaction Evaluation Scale (HRIES [7]), is adopted.

We choose the HRIES scale because, unlike the state of the art [31, 149], the authors in [7] developed the HRIES scale using different types of robots (not just humanoids), making the scale *generalizable* to all types of robots. Additionally, unlike other scales that validate the questionnaire with static images [149], the authors in [7] validate the scale by conducting tests with robots in motion (via videos). Furthermore, the authors in [7] also conduct a test with a real robot to further assess the validity of the scale.

Since, for conducting the real-world experiment, we use a non-humanoid robot that navigates and shares the environment with humans, the aforementioned scale is particularly well-suited for our case.

Figure 6.9 illustrates the scale developed in [7]. The scale is designed to assess how humans perceive robots, with a particular focus on *anthropomorphism*. However, the concept of anthropomorphism is too broad to be measured directly. Therefore, the authors identify four *factors* (shown on the right in Figure 6.9) that are measured through what they refer as *items* (depicted on the left in Figure 6.9).

The *factors* on the scale are: sociability, animacy, agency, and disturbance. Sociability is characterized by the capacity of an individual or group to interact effectively and with positive engagement with others. The animacy suggests human characteristics for non-human agents. The agency is the capacity of a robot to act independently and make its own decisions, thus is strictly linked with the perceived intelligence of the robot. The disturbance captures the negative valence associated with perceptions of robot, including feelings of discomfort and specific negative evaluations.

The questionnaire is available in English and Italian, allowing participants to choose the language they feel most comfortable responding in. The Italian translation of the HRIES scale is presented in Table 6.1. The translation of the *items* is conducted with the assistance of the scale author, Dr. Spatola.

Regarding the practical use of the scale, we follow the instructions reported in [7], tailoring the research question to align with the robotic behavior of interest: the robot's movement. In particular, each item is rated using a 7-point Likert scale, answering the following question:

"Using the provided scale, how closely are the words below associated with the robot's motion during the experiment?"

By collecting data on the anthropomorphization of the robot's behavior, we collect information about how humans perceive the robot. Specifically, the algorithm that is perceived with the highest anthropomorphism probably shares more human-like characteristics than the other algorithms, being the most socially accepted [33].

Items	Factor
Warm	Sociability
Likeable	Sociability
Trustworthy	Sociability
Friendly	Sociability
Alive	Animacy
Natural	Animacy
Real	Animacy
Human-like	Animacy
Self-reliant	Agency
Rational	Agency
Intentional	Agency
Intelligent	Agency
Creepy	Disturbance
Scary	Disturbance
Uncanny	Disturbance
Weird	Disturbance

Fig. 6.9 The Human-Robot Interaction Evaluation Scale (HRIES) [7].

Items in English	Items in Italian
Warm	Caloroso
Likeable	Piacevole
Trustworthy	Affidabile
Friendly	Amichevole
Alive	Vivo
Natural	Naturale
Real	Reale
Human-like	Sembianze umane
Self-reliant	Autonomo
Rational	Razionale
Intentional	Intenzionale
Intelligent	Intelligente
Creepy	Raccapricciante
Scary	Spaventoso
Uncanny	Inquietante
Weird	Strano

Table 6.1 Italian translation of the items in HRIES.

6.3 Experimental setup

The room of the real-world experiment is approximately 40 m^2. The involved participants are 2 and one robot (Locobot WX250), as shown in Figure 6.10. Within the experiment, every participant is assigned a specific starting point and a corresponding goal to achieve.

Fig. 6.10 Scenario real-world experiment with two participants and the robot.

The experimental protocol comprises **three phases**:

1. **Explanation of the experiment and collection of participant general information**.

   In this phase, each participant receives detailed information regarding the research objectives, experimental procedure, risks and benefits of the experiment, and the right to withdraw at any time from the experiment. Additionally, each participant is given a consent form to sign, granting their participation in the research project. Then, the participants are required to provide their gender, age, and level of experience in the robotics field on a Likert scale from 1 (no experience) to 5 (expert). This phase is expected to take about 6 minutes;

2. **Participant training**.

   During this phase, participants are allowed to walk in the room for 3 minutes without the robot. Subsequently, participants have another 3 minutes to walk in the same room but with the robot. This phase is essential to allow each participant to get familiar with the environment and the experiment scenario;

3. **Experiment**.

   In this phase, participants walk from a predefined starting point to a goal. This phase will consist of *3 sessions*, one for each algorithm. Within each session, *4 trials* are conducted. At the end of each session, each participant completes an online questionnaire based on the HRIES scale [7].

The order of the sessions for each experiment is randomized to avoid collecting data with biases related to the order of execution of the different algorithms. An example of session randomization is described in Figure 6.11. Each session is expected to take about 5 minutes. To complete the entire experiment, each participant needs approximately **30 minutes**.

	Session 1		Session 2		Session 3	
1st experiment	SFM 4 times	Questionnaire	GTSFM 4 times	Questionnaire	ORCA 4 times	Questionnaire
2nd experiment	GTSFM 4 times	Questionnaire	ORCA 4 times	Questionnaire	SFM 4 times	Questionnaire
3rd experiment	ORCA 4 times	Questionnaire	GTSFM 4 times	Questionnaire	SFM 4 times	Questionnaire
4th experiment	GTSFM 4 times	Questionnaire	SFM 4 times	Questionnaire	ORCA 4 times	Questionnaire
5th experiment	SFM 4 times	Questionnaire	ORCA 4 times	Questionnaire	GTSFM 4 times	Questionnaire
6th experiment	ORCA 4 times	Questionnaire	SFM 4 times	Questionnaire	GTSFM 4 times	Questionnaire

Fig. 6.11 Randomization of test sessions across different experiments.

6.4 A-priori power analysis

Our study aimed to investigate the subjective perception of the GTSFM algorithm by comparing it with SFM and ORCA algorithms. To estimate the number of participants ensuring the reliability and validity of our findings [95], we conducted a rigorous a-priori power analysis, employing the freely available software G*Power [96]. The HRIES questionnaire, chosen in our study for the real-world experiment, was validated through different studies in [7], where a normal distribution of data was assumed. Based on these previous studies, we also assumed that our data adhere to a normal distribution. Consequently, this allows us to confidently support the use of parametric tests in our analysis [150]. Then, we assumed that the data collected during the real-world experiment would be analyzed via the parametric one-way ANOVA because our independent variables were more than two independent groups (GTSFM, SFM and ORCA) and our dependent variables (the factors of the questionnaire) were continuous since the factors were computed through the Principal Component Analysis [151]. The result of the a-priori analysis for the one-way ANOVA is 66 volunteers, considering an alpha level equal to 5%, power of the study 80% and the three groups, corresponding to the three different algorithms. We recruited the participants using the Institutional email of Politecnico di Torino

and then, we invited them to come to the lab, where the experiment took place. Ultimately, we collected 76 responses, exceeding the sample size of 66.

6.5 Statistical analysis of data

The objective of the statistical data analysis of the *qualitative* data is to reject the following null hypothesis:

All algorithms (GTSFM, SFM, ORCA) are perceived by participants as indistinguishable.

To achieve this objective, first of all, it is necessary to reduce the dimension of the dataset from the items to the possible factors through the principal component analysis (PCA) [151]. Before performing the PCA analysis, the merging of the data collected for each algorithm is necessary, since the PCA requires a single dataset of all the data collected. Then, to ensure that all items have the same scale, the standardization of the data is performed. Subsequently, the PCA is performed and the loadings are rotated. The rotation of the loadings is a post-processing step that can improve the interpretability of the results by making the loadings more meaningful [151]. Finally, the *score* of each principal component for each participant is computed, considering the linear combination of the standardized data with the computed loadings.

Then, to reject the null hypothesis, the one-way ANOVA is performed on the computed *score*. If the one-way ANOVA test produces a statistical significance value (p-value) below 5%, it means there is the presence of at least one group that has a significant difference from the others in terms of some factor. To discover which groups are classified as significantly different, a post-hoc analysis is performed considering two pairs of groups at a time.

6.6 Results

We collected 76 responses for the real-world experiment, where the average age of the participants was approximately 25, having limited robotics experience. The gender composition was unbalanced toward men (Table 6.2).

Number of Participants	76
Gender	Male 73.7% and Female 26.3%
Age	24.99 ± 3.91
Experience with robotics	2.03 ± 1.07

Table 6.2 Demographic characteristics and participants' levels of experience with robotics, assessed on a scale ranging from 1 (indicating minimal experience) to 5 (indicating maximal experience). This data is gathered in the first phase of the experiment.

Firstly, experimental outcomes were analyzed with the PCA to reduce the dimension of the dataset through the identification of the latent factors. Then, the one-way ANOVA was performed to statistically reject the following null hypothesis.

Our null hypothesis posits: *All algorithms (GTSFM, SFM, ORCA) are perceived by participants as indistinguishable.*

A summary of the PCA loadings for each item is presented in Figure 6.12. The latter highlights the four main components that exhibit the highest variances. We focused on these four components because we wanted to interpret each component with the factor of the HRIES scale. However, as can be easily deduced from the figure, only the first two components can be directly associated with two factors of the scale. The remaining two factors do not align significantly with any of the PCA components. In particular, the first component exhibits a significant concentration of items associated with *agency*, while also displaying a moderate influence from *animacy* and *sociability*. The second component predominantly comprises items associated with *disturbance*. Nevertheless, the lack of interpretability for the remaining two components associated with *sociability* and *animacy* factors might be attributed to the non-human appearance of the robot, which precludes reliable assessments for these factors.

Mean and standard deviation of component scores are presented in Figure 6.13. Specifically, the measure of the perceived *agency* across the three algorithms is illustrated in Figure 6.13a, where the SFM is characterized by the highest average score, followed by GTSFM and ORCA. However, despite the initial indications of the one-way ANOVA ($F = 4.54$, $p = 0.01$) of significant differences in agency scores among the algorithms, a post-hoc analysis (Table 6.3) revealed that only the pair SFM-ORCA was statistically different. Participants perceived the remaining pairs as being indistinguishable.

Latent factor	Item	1st Component	2nd Component	3rd Component	4th Component
Sociability	Warm	-0,0588	**0,0200**	-0,5046	-0,0033
Sociability	Likeable	**0,0105**	-0,0224	-0,2807	-0,2751
Sociability	Trustworthy	**0,2358**	-0,1018	-0,1567	-0,0793
Sociability	Friendly	0,0561	-0,1098	-0,5934	**0,1916**
Animacy	Alive	**0,1870**	0,1652	-0,2116	-0,1057
Animacy	Natural	-0,0224	**0,0741**	-0,1025	-0,4938
Animacy	Real	**0,2165**	0,0325	-0,0163	-0,2919
Animacy	Human-like	-0,0359	**0,0642**	-0,3922	-0,1257
Agency	Self-Reliant	**0,6017**	-0,0831	-0,0042	0,2245
Agency	Rational	**0,4628**	-0,0189	0,0080	-0,0032
Agency	Intentional	**0,4327**	0,0812	0,1757	-0,1421
Agency	Intelligent	**0,2469**	0,0565	-0,0083	-0,3085
Disturbance	Creepy	-0,0589	**0,5444**	0,0289	-0,0125
Disturbance	Scary	0,0562	**0,5741**	0,1091	-0,0366
Disturbance	Uncanny	-0,0477	**0,4997**	-0,1532	0,0520
Disturbance	Weird	0,1441	0,2145	-0,1157	**0,5961**
Percentage of explained variance		41,33%	16,95%	7,18%	4,89%

Fig. 6.12 Loadings for each item considering 4 principal components as suggested by the HRIES scale [7]. In bold is highlighted the maximum value for each item. Highlighted in yellow are the items that make the most significant contributions to that component.

On the other hand, the measure of the perceived *disturbance* across the same algorithms is illustrated in Figure 6.13b, where ORCA is characterized by the highest average score, followed by GTSFM and SFM. Nevertheless, the one-way ANOVA, conducted on disturbance scores, reveal the absence of statistically significant differences ($F = 0.67$, $p = 0.51$). These findings indicate that participants were unable to detect differences in comfort between the three algorithms.

Algorithm 1	Algorithm 2	p-value
SFM	ORCA	0.009
SFM	GTSFM	0.119
ORCA	GTSFM	0.586

Table 6.3 Post-Hoc analysis on the score of the 1st component. In yellow is highlighted the statistically different pair.

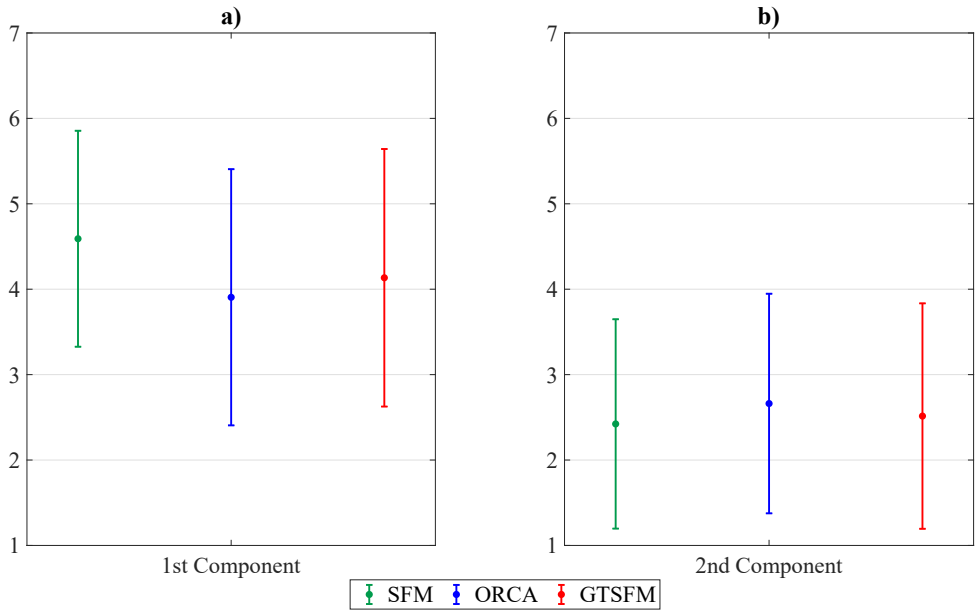


Fig. 6.13 Mean and standard deviation of component scores across different algorithms: SFM (Social Force Model), ORCA (Optimal Reciprocal Collision Avoidance), GTSFM (Game-theoretic Social Force Model). The components are: a) 1st Component is the perceived *agency*, b) 2nd Component is the perceived *disturbance*.

6.7 Discussion

This study aims to develop a navigation system that generates trajectories for autonomous robots to ensure the social acceptance of humans by leveraging the concept of anthropomorphism. To achieve this goal, we employed the social force model combined with game theory (GTSFM) to model human motion intentions and devised a trajectory planning algorithm based on this model.

To assess the social acceptability of the generated robotic trajectories, we conducted a twofold validation: first, we evaluated *quantitatively* the performance of the generated trajectories across three experimental conditions (SFM, ORCA and GTSFM) through four performance parameters of the state of the art (PLR, AS, CPD, and PR). This validation was done through a Monte Carlo simulation campaign. Then, we performed a *qualitative* analysis through a survey questionnaire with a statistically significant group of volunteers using the HRIES scale [7]. This validation was done through a real-world experiment.

The most significant finding of the *quantitative* analysis is that the GTSFM algorithm produces smoother paths than the two state-of-the-art algorithms, resulting in a more natural motion. Moreover, through numerical validation, we have consistently observed that GTSFM outperforms SFM across the four performance metrics. In particular, the results suggest that GTSFM excels in finding shorter, smoother paths with a higher speed than SFM, reducing the time to reach the goal while maintaining a safer distance from humans. This finding highlights the effectiveness of GTSFM in achieving more efficient and natural navigation compared to SFM.

While ORCA excels in speed (AS) and efficiency (PLR), its social navigation capabilities are lower than GTSFM. Indeed, ORCA's trajectories tend to be less smooth and closer to pedestrians, potentially increasing the risk of collisions and pedestrian disturbance.

Nevertheless, these *quantitative* findings are not consistent with the *qualitative* results. The latter revealed that none of the considered principal components showed a statistically significant difference between the algorithms, except for SFM and ORCA in the case of agency (the first component). This apparent contradiction between the two types of analysis raises interesting questions about the reasons behind these findings. This discrepancy could potentially stem from the influence of unaccounted factors. One such factor might be the robot's appearance, which obscures the distinction between the algorithms. Additionally, the limited velocity of the real robot (0.5 m/s) compared to the average human walking velocity (1.4 m/s [152]) might have hindered the identification of a potentially anthropomorphic algorithm.

Algorithm acceptability is evaluated through both *quantitative* and *qualitative* measures, each addressing distinct aspects. Nonetheless, some measures aim to understand shared underlying factors, allowing for a unified discussion of these measures.

Quantitative measures, such as PLR, AS, CPD, and PR, provide objective assessments of algorithm performance. PLR indicates the overall efficiency of the algorithm, while AS and CPD measure how comfortable the generated trajectories are for humans. In particular, the human comfort decreases as the robot's speed of movement increases [83]. Instead, the human comfort increases as the robot's distance from humans increases [83]. PR assesses the smoothness of the trajectories, which is related to the naturalness of the robot's movements [27].

On the other hand, the *qualitative* measurement based on the HRIES scale, focuses on the anthropomorphism of the robot's motion. These measures include sociability, animacy, agency, and disturbance.

Building upon this explanation, further insight could be gained by bridging the gap between *quantitative* and *qualitative* results. This could be achieved by analyzing the relationship between objective measurements like AS and CPD with the subjective evaluation of perceived *disturbance* captured by the HRIES scale.

Considering the AS, the ORCA algorithm has the highest average speed value which turns out to be the one with the highest perceived disturbance among the three algorithms, although there is no statistical difference across the considered algorithms. This statistical indistinguishability between the algorithms may be attributed to the limitations of the robot's velocity ($0.5\ m/s$). While the ORCA algorithm might theoretically promote faster movements compared to other algorithms, the robot's hardware limitations restrict its maximum achievable velocity. This velocity may be insufficient to elicit feelings of discomfort in human subjects which may lead to the absence of an identification of a statistically significant difference between the algorithms.

While the GTSFM exhibited a greater average CPD compared to the SFM, real-world experiments did not translate this objective metric into a higher perceived comfort for GTSFM than SFM. This discrepancy can be explained by considering the different contexts in which the measurements were taken. *Quantitative* measurements were conducted in a simulated environment, where pedestrians exhibited rational movements and interacted with the robot as another human in a shared space. In contrast, the real-world experiment showed that despite the robot being programmed to maintain a specific distance, participants tended to approach closer than they would with a real person (maybe for curiosity reasons), thereby obscuring the distinctions between the different algorithms.

Another potential comparison between *quantitative* and *qualitative* analyses could be made by correlating PR measurements with the evaluation of *animacy*. This comparison is justified because when a robot exhibits smooth movements, resembling the typical non-nervous motions of humans. Thus, the robot is perceived as more human-like [27]. Unfortunately, this comparison cannot be made since after the PCA, the *animacy* cannot be uniquely attributed to any of the identified components, thus failing to be interpreted and discussed. We encountered a similar challenge regarding *sociability*, as we could not unequivocally attribute a principal component to this factor.

We have reason to believe that probably these two problems are caused by the robot's appearance. Although the questions were about the robot's movement, it is impossible to completely separate the robot's movement from its appearance. This makes the measures of *animacy* and *sociability* unreliable for further discussion.

These findings are consistent with other studies in literature [153, 33], where the authors have shown that the physical appearance of a robot plays a significant role in shaping how people perceive and interact with it. This highlights the importance of considering the appearance of a robot when developing a socially-aware robot algorithm. Moreover, in [154], authors have investigated how the appearance and movement characteristics of a robot can influence people's perceptions of its animacy. The main result of the study in [154] is: when the robot has a human-like appearance, humans perceive naturalistic motion as more animate compared to mechanical motion. This difference is not perceived when the robot lacks human-like appearance. This finding aligns with our results on animacy, where we observed *quantitative* differences between algorithms but no *qualitative* distinctions.

When interpreting the results of our study, we should also acknowledge the limitations of the *quantitative* and *qualitative* validation. Regarding the former, it was conducted in a simulated environment where humans were not real people but rather controlled by a human motion model. This model was used to simulate various human behaviors by adjusting different parameters. However, this approach has two limitations: first, it does not explicitly model the stochastic nature of human behavior, even though this is implicitly accounted for through parameter tuning. Second, it is not possible to accurately assess real human-robot interaction, as humans simply avoid the robot as another peer human in the simulation.

On the other hand, one of the main limitations of the *qualitative* validation lies in the constrained movement speed imposed upon the participants. This artificial restriction was necessary to ensure interaction with the robot, as its maximum speed is 0.5 m/s. Furthermore, such restricted speed of the robot may have limited the range of conditions tested among the different experimental conditions and consequently, the ability to comprehensively assess perceived comfort in human subjects.

The non-humanoid appearance of the robot and its relatively small size likely contributed to limitations in measuring sociability and animacy. Moreover, during the experiment, participants did not engage in a peer-to-peer interaction with the robot, instead assuming a leader-follower interaction. This phenomenon could be attributed to the robot's non-humanoid characteristics, such as the absence of a face, two arms and a typical body structure. Additionally, the robot's height of 63 cm, significantly below the Italian average height of humans (1.68 m [155]), contributes to an appearance reminiscent of a toy-like aesthetic. This lack of human-like features may have hindered the development of a peer-to-peer rapport.

Throughout the experiment, participants consistently approached the robot more closely than they did other humans. This behavior may be attributed to either the participants' natural curiosity or the perceived non-threatening nature of the robot, possibly influenced by its small size.

Our work can be extended in several directions.

To make the *quantitative* analysis as realistic and precise as possible, a real experiment should be created by placing the robot in a real environment (such as a corridor in a hospital or at university). In this way, we would have real people interacting with the robot, rather than agents simulated by models. In this ideal test, reliable data could also be collected about human-robot interaction (as people would feel free without the constraints and biases of laboratory experiments). This approach allows us to collect more realistic data than our simulated data. Furthermore, this approach could help to understand the type of interaction between humans and robot, whether it is a peer-to-peer or leader-follower interaction.

Regarding the *qualitative* analysis, we believe that the experiment protocol is reliable and the biggest limitations were the limited robot's speed and its appearance. Therefore, we recommend repeating the experiment proposed in this study, choosing a robot that has a speed comparable to humans (1.4 m/s [152]). In addition, the author in [33] showed that in evaluating how humanlike a robot appears, the robot's

head and face receive considerable attention, since this body part is crucial in human-human communication. Therefore, we recommend a robot that has this body part. The choice of the robot's height is also essential [156]. For this reason, a robot with a height very similar to humans (1.68 m) should be chosen.

The main result of this study is that probably the choice of the robot is essential to ensure an interaction as similar as possible to that of humans to obtain statistically significant results even at the level of motion measurement.

Part II

Part 2 Mission Planning with Human Supervision

Chapter 7

Auction-based task allocation and motion planning for multi-robot systems with human supervision

This chapter presents a task allocation strategy for a multi-robot system with a human supervisor in a dynamic scenario. The human supervisor can intervene in the operation scenario by approving the final plan before its execution or forcing a robot to execute a specific task. The proposed task allocation strategy leverages an auction-based method in combination with multi-goal motion planning. The latter is used to evaluate the costs of execution of tasks based on realistic features of paths. An extensive simulation campaign in a rescue scenario validates our approach in dynamic scenarios comprising a sensor failure of a robot, a total failure of a robot, and a human-driven re-allocation. We highlight the benefits of the proposed multi-goal strategy by comparing it with single-goal motion planning strategies at the state of the art. Finally, we provide evidence for the system's efficiency by demonstrating the powerful synergistic combination of the auction-based allocation and the multi-goal motion planning approach. At the end of this chapter, a discussion of the results is provided.

7.1 Background

In the near future, multi-robot systems (MRS) are envisaged to significantly impact different social fields [157]; from surveillance missions [39] to industrial applications [158], from rescue operations [34] to agriculture [38].

MRSs exhibit significant advantages over Single-Robot Systems (SRS), due to their redundancy, flexibility, efficiency, and the absence of a single point of failure [159, 160]. However, communication, coordination, and control overhead are required in order to orchestrate the action of the team as a whole.

Complex and life-critical tasks, such as rescue operations, often involve the adoption of MRSs consisting of unmanned vehicles and human operators, where humans are in charge of important decisions and of some aspects of the coordination of the operations, especially those related to the evaluation of the overall success of the work plan, the safety of human lives, and the management of unforeseen situations. Rescue operations, in particular, involve a large quota of human operators within the team, which may attain up to two humans for each robot [161].

Reducing the number of human operators in such teams is desirable to enhance safety, avoid confounding factors emerging from the adoption of contrasting strategies by different operators in the team, and reduce the odds of human mistakes [162]. On the other hand, operations that involve ethical challenges related to *decision-making* [163] and *responsibility* on decision [164] will continue to require human intervention or supervision in the unforeseeable future [13].

For example, choices about allocating resources in emergency situations, including where to concentrate rescue efforts, assessing risks, determining the order of people to be rescued, prioritizing medical treatment, managing who must be left to wait, and optimizing the utilization of scarce resources are unplausible to be made by teams constituted by robots only [163]. In this context, Harbers et al. [164] raise the issue related to moral and legal responsibility, where the former concerns blame and the latter concerns accountability. These issues, according to the authors, occur when robots are not supervised by a human. If a robot undergoes a malfunctioning, behaves inappropriately, makes an error, or causes harm, it can be difficult to determine who is responsible for the resulting damage. This issue becomes even more complex when the robot has some level of autonomy, self-learning abilities, or is capable of making decisions that were not explicitly programmed.

This calls for the design of robust and efficient mission planning to coordinate the MRSs with human supervisors, referred as to HMRS in the following. Our mission planning framework comprises two key components: task allocation and motion planning, facilitating the efficient calculation of task costs.

In this work, we propose a task allocator strategy for a HMRS with heterogeneous capabilities, where the human supervisor can be either a pilot of one of the robots or an external coordinator. The proposed task allocation can manage a dynamic environment, involving both changes in the operation requirements or in the robots' capabilities. Also, a human supervisor can intervene in the planning process by: *(i)* approving or canceling a proposed plan; or *(ii)* introducing new constraints to a proposed plan; for example, by assigning a specific task to a given robot, along with a given execution time set for safety reasons or due to a change in the capabilities required to execute a given task. We advance the state of the art along two main directions. First, our task allocation combines an auction-based strategy [165] with a motion planner [166] enhanced with a multi-goal approach, to take full advantage of the features of the sequential single-item auction and leverage real and measurable features of the path of to be accomplished, rather than its mere description. Second, flexibility in operations is attained through dynamic re-allocation, which can be triggered at any time, either by changes in the operational conditions or by the human supervisor.

7.1.1 Previous works

Our contribution falls in the broad category of multi-robot task allocation problems (MRTA) [167] –a variant of the multiple Traveling Salesman Problem (mTSP) [167], which is notoriously NP-hard.

Main approaches to the solution of task allocation problems are Mixed-Integer Linear Programming (MILP) [168, 40], and auction-based techniques [39, 165, 41, 169, 42, 43, 170].

The former may lead toward the optimal solution at the cost of an often unaffordable computational complexity, which calls for the combined usage of heuristics and the consequent attainment of suboptimal solutions. In the context of HMRS, the support of a human supervisor was included in [40] to evaluate the intermediate solutions of the MILP based on objective or subjective quality criteria and personal

expertise. In this way, also sub-optimal solutions may be adopted, and the solver can be conducted to an early termination. However, in this case, the human supervisor is continuously required to evaluate operational scenarios, practically providing heuristic criteria to reduce the computational load of the solver, at the expense of their own cognitive load, entailing an increase in the level of stress and subtracting precious intellectual resources to the execution of complex tasks.

The latter, on the other hand, consists of an iterative strategy based on the optimization of the interest of selfish agents, typically leading to sub-optimal solutions with a reasonable computational complexity. Auction algorithms have grown in popularity within the robotics community [165] to handle task allocation problems [39, 171] efficiently and robustly [41]. In an auction, each robot (the bidder) places a bid to commit to the execution of each task (item) based on a given cost function. Then, a coordinator (the auctioneer) assigns (sells) the items to the highest or the lowest bidder, depending on whether the considered cost function should be maximized or minimized [165]. Auctions are particularly suitable for dynamically changing environments and can be deployed in centralized, decentralized, and distributed architectures [172]. Specifically, the calculations of the auctioneer and the bidders can be done on a single system (centralized), multiple systems (decentralized), or without a unique and centralized auctioneer (distributed).

In the literature, auction-based methodologies have been used in different applications with MRS, such as exploration and destruction, patrolling, and surveillance mission, [39, 41, 44]. Notably, [39] adopts an auction-based methodology to solve a task allocation problem of a HMRS in a dynamic scenario with priority constraints between tasks. Differently from our approach, however, the human-controlled vehicle has neither supervisory features nor specific privileges. This makes the solution of the problem equivalent to that computed for a fully automated team.

Some works in the literature aimed at combining a task allocation strategy with a motion planning. In [173] a MILP is combined with an RRT*-based algorithm, while in [174] an integer programming model integrates a motion planner based on a genetic algorithm. Instead, authors in [43, 175] integrate an auction-based task allocator with the A* algorithm. Specifically, in [43], the authors apply the auction in a dynamic environment for UAVs where the mission is continuously allocated and executed autonomously.

Auction-based task allocation is also combined with RRT-based algorithms [176, 177]. RRTs are suitable for supporting task allocation because they can rapidly compute a path in the search space by constructing an incremental exploration tree [178]. However, studies in [176, 177] exploit the standard RRT algorithm, which has the drawback of computing non-optimal solutions. The optimality of the motion planning is an essential feature for the quality of the task allocation because the computed paths are evaluated to assign the task to the robot that offers the best solution. For this reason, differently from [176, 177], our motion planner guarantees an optimal path thanks to the $RRT^{\#}$ algorithm [166].

7.1.2 Our contributions

The HMRS aims to handle a complex operation happening in a dynamic environment. We assume that such an operation may be decomposed according to a hierarchical structure, illustrated in Figure 7.1.

Such a complex operation consists of some independent sub-operations that must be executed by a robot with appropriate capabilities. Each sub-operation may have a priority. Each sub-operation is in turn composed of several tasks, subject to precedence constraints.

The operation structure mentioned above is relevant to many complex operation scenarios, such as people rescuing. In this context, the operation consists of some sub-operations equal to the number of people to be rescued. In particular, each sub-operation consists of all the actions (tasks) necessary to save one person (target). Each target has a priority that is related to the urgency of the rescue. Each task coincides with the visit of a location in the operational scenario.

The HMRS consists of a team of heterogeneous robots, i.e. each robot has a set of capabilities, which allow it to execute certain tasks. The dynamic nature of the environment where the HMRS operates may elicit re-allocation upon changes in the operational conditions, also called *perturbations.* For instance, a robot or one of its capabilities may become unavailable due to a collision, a system failure, the exhaustion of its battery; or the human supervisor demands a re-allocation, due to safety reasons or other technical considerations not intelligible by machines. More specifically, the human supervisor may trigger a re-allocation of the HMRS through one of the following actions: *(i)* rejecting the computed plan before its execution;

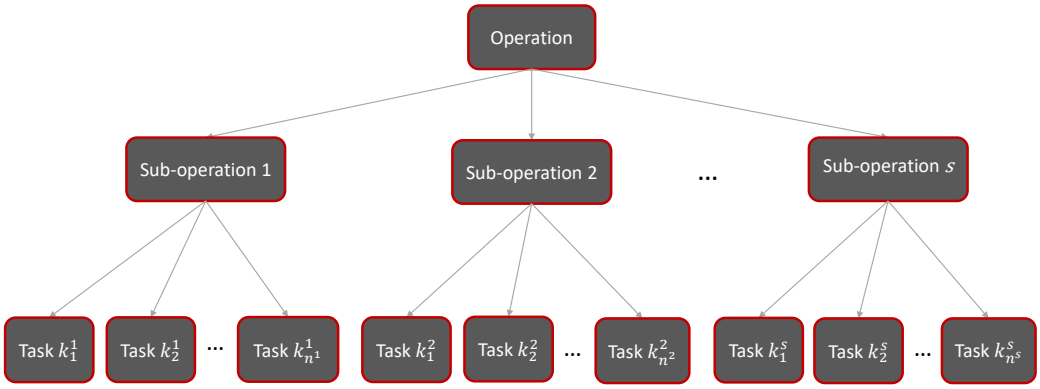


Fig. 7.1 Hierarchical structure of operation, sub-operations, and tasks.

or, *(ii)* forcing a particular robot to execute a task. Regarding the rescue operation, the former is typically related to an overall approval of the plan, in light of ethical or safety implications, while the latter may relate to on-the-go decisions that may increase the chances of safety in light of the actual operational conditions. Such a dynamic scenario is summarized in the flow chart of Figure 7.2.

Our effort presents several novelties and improvements compared to the state of the art. First, we propose an auction-based method for a heterogeneous team operating in a dynamic scenario with human supervision, which supports precedence constraints between tasks, priority between sub-operations, and on-the-go re-allocation due to perturbations, coming either from the environment or from the human supervisor –a setting that was not entirely contemplated in the past [39–44]. Along the lines of [40, 179], we design a system able to simulate the intervention of the human supervisor by dynamically adding constraints to the auction-based task allocation, such as forcing a robot to execute a specific task within a given completion time or changing the capability required for a given task. This scenario may arise for safety reasons or, for example, to adjust the allocation problem when a malfunction occurs, thus enabling the operation to be completed. However, the MILP approach used in [40] hampers its concrete applicability to complex and dynamically changing scenarios, due to its inherent computational burden. Conversely, Hussaini et al in [179] describe a scenario where the multi-robot system is supervised by a human operator who can actively address corrective actions in the assignment plan based on the estimated or the notified contingencies. However, their re-allocation process is handled by using a heuristic-based task allocation, which may face scalability issues,

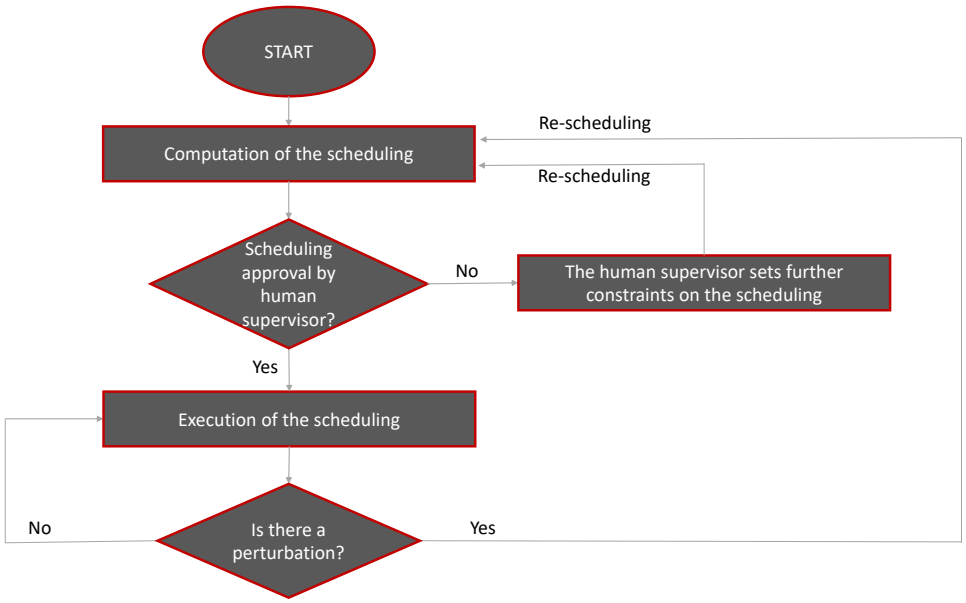


Fig. 7.2 Dynamic scenario of operations and re-allocation mechanisms.

making it inefficient or impractical to find a suitable allocation within a reasonable time.

Second, along the lines of [180], we use a multi-goal motion planner in combination with the auction-based allocation, achieving an overall method that is fast, effective, and reactive to perturbations. However, in [180], the authors are focused more on the optimality of the planning rather than on the responsiveness of the whole system. Also, the efficiency of the strategy claimed in [180] is hampered by the assignment of capacity constraints to each robot. Notably, besides six scheduled tasks for each robot, the computational burden tends to become unmanageable. Here, such constraints are not posed and efficiency is privileged, if necessary, through a trade-off between computational burden and pursuit of optimality. Moreover, in [180] the authors use the general logic of the auction but the motion planner does not leverage any particular feature of the auction to work in synergy with it. In fact, the motion planner creates a general graph that is used to evaluate the effect of the candidate task on the entire robot schedule. In particular, the graph is useful to evaluate different links (robot-task and task-task) and finally to compute the best solution for the TSP problem. Here, on the other hand, the multi-goal motion planner is used to fully leverage the features of the sequential single-item auction by simultaneously computing the cost for each robot to accomplish a given task.

To the best of our knowledge, there are no works that describe a system incorporating a human supervisor with the aforementioned functions, imposing constraints on an auction-based task allocator with the described features, specifically designed to address ethical challenges in demanding environments.

Moreover, we remark that, although the strategy proposed in this thesis is tailored to a rescue scenario for illustrative purposes, the application field is extensive and may embrace robots of heterogeneous nature, such as ground, aerial, or underwater.

The rest of the chapter is structured as follows. Section 7.2 explains the problem statement and states the assumptions of our approach. Section 7.3 presents our methodology, based on an auction-based task allocator and a multi-goal RRT$^{\#}$ algorithm. The effectiveness and robustness of our methodology are demonstrated by simulations in Section 7.4. Finally, in Section 7.5, we draw our conclusions and offer a discussion toward further developments.

7.2 Problem statement

In the following, we use roman font to denote scalar quantities ($x \in \mathbb{R}$), low bold font to denote vectors ($\boldsymbol{x} \in \mathbb{R}^2$), and upper bold font to denote set ($\boldsymbol{X} \in \mathbb{R}^N$) and matrices ($\boldsymbol{X} \in \mathbb{R}^{N \times M}$).

We assume a two-dimensional operational space $\boldsymbol{X} \in \mathbb{R}^2$ defined as a Euclidean state space in which each element $\boldsymbol{x} \in \boldsymbol{X}$ represents a possible location for a robot. The subset $\boldsymbol{X}_{\text{obs}} \subseteq \boldsymbol{X}$ contains locations where a robot cannot be located, e.g. those occupied by obstacles. We assume that the positions of the obstacles are known a-priori to the task allocator and the motion planner. $\boldsymbol{X}_{\text{free}} = \boldsymbol{X} \setminus \boldsymbol{X}_{\text{obs}}$ includes the remaining positions where a robot can be located, also called the *valid* locations.

The HMRS comprises m robots and is identified by the set $\boldsymbol{R} = \{r_1, r_2, \ldots, r_m\}$. The set $\boldsymbol{X}_{\text{r}} = \{\boldsymbol{x}(r_1), \boldsymbol{x}(r_2), \ldots, \boldsymbol{x}(r_m)\}$ indicates the position of each robot, with $\boldsymbol{x}(r_i) \in \boldsymbol{X}_{\text{free}}$, $i = 1, \ldots, m$.

The set $\boldsymbol{Cap} = \{p_1, p_2, \ldots, p_l\}$ indicates the l available capabilities used to execute all the tasks by the multi-robot system. A capability is a particular feature that empowers a robot to accomplish a particular operation; for example, the capability of moving hazardous materials or to illuminate the scene at night.

Each robot has different capabilities that may change over time. They are summarized in a boolean time-varying matrix $\boldsymbol{RC(t)}$ of dimensions $m \times l$. The element $RC(t)_{i,j}$ is set to one if the robot i, with $i = 1,\ldots,m$, is equipped with the capability j, with $j = 1,\ldots,l$, and to zero otherwise.

The operation to be allocated aims to manage s targets defined by the set $\boldsymbol{G} = \{g_1, g_2, ..., g_s\}$. The set $\boldsymbol{X}_g = \{\boldsymbol{x}(g_1), \boldsymbol{x}(g_2), \ldots, \boldsymbol{x}(g_s)\}$ indicates the position of each target $g \in \boldsymbol{X}_{free}$. Each target has a priority defined by the set $\boldsymbol{GP} = \{gp_1, gp_2, ..., gp_s\}$ that defines which target has to be managed first, with $gp_i \in \mathbb{N}$, $i = 1,\ldots,s$.

In particular, the higher the priority, the more urgent the target to manage. Nevertheless, it might also happen that two or more sub-operations have the same priority, then the auction will try to handle them in parallel, when possible.

Hence, the operation consists of s sub-operations because each sub-operation is responsible for managing only one target while respecting its priorities.

Each sub-operation consists of several tasks. The set $\boldsymbol{K^i} = \{k^i_1, k^i_2, \ldots, k^i_{n^i}\}$ denotes the list of n^i tasks that form the sub-operation i, with $i = 1, 2, ..., s$, in which the subscript represents the sequencing of the tasks. Tasks must be performed sequentially. For instance, task k^i_2 has to be performed after the task k^i_1. $\boldsymbol{K}^{tot} = \boldsymbol{K^1} \cup \boldsymbol{K^2} \cup \ldots \cup \boldsymbol{K^s}$ represents the set of all tasks, with cardinality $n^{tot} = n^1 + n^2 + ... + n^s$.

Each task has to be performed in a specific location. The set $\boldsymbol{X}_k \subseteq \boldsymbol{X}_{free}$ includes the positions of the free space, where all tasks must be executed. The notation $\boldsymbol{x}(k^i_j) \in \boldsymbol{X}_k \subseteq \boldsymbol{X}_{free}$ indicates the position of a task k_j of the sub-operation $i = 1, ..., s$. We assume that the task allocator and the centralized motion planner know the positions of every robot $\boldsymbol{x}(r_i) \in \boldsymbol{X}_r$ and every task $\boldsymbol{x}(k^i_j) \in \boldsymbol{X}_k$.

The subdivision of the operation in sub-operations and, subsequently, in tasks is shown in Figure 7.1. The decomposition of the complex operation in its tasks is out of the scope of this thesis; hence, we assume that the sets of sub-operations and tasks are made available to the task allocator by an external mechanism.

Each task requires some capabilities to be performed. The combination of tasks and capabilities is summarized in a boolean matrix $\boldsymbol{TC}$ of dimensions $n^{tot} \times l$. Element $TC_{i,j}$ is set to one if the task i, with $i = 1, 2, \ldots, n^{tot}$ requires the capability j, with $j = 1, 2, \ldots, l$; it is set to zero otherwise.

The role of the task allocation is to handle the n^{tot} tasks to the HMRS composed by m robots equipped with different capabilities l. The computed plan is designed

to optimize the total time of the operation, guaranteeing that tasks are executed by the robots that possess proper capabilities, respecting the prioritization between sub-operations and precedence constraints between tasks. Re-allocation can be triggered by *perturbations,* which can be *external* or *internal* .

An *external* perturbation is caused by an external and unexpected event, such as a system or sensor failure which can cause the loss of a robot or the loss of its capabilities.

An *internal* perturbation occurs when it is caused by an internal event, e.g. a change of strategy forced by the human supervisor, such as changing capability for a given task or assigning a task to a particular robot.

In the following, the term t_{new} defines the time instant when a perturbation occurs considering a continuous time.

The intervention of the human supervisor is defined by the boolean matrix $\boldsymbol{TC}^{\text{H}}$ of dimension $n^{\text{tot}} \times l$, in which each element $TC^{\text{H}}_{i,j}$ defines if the human supervisor forces the capability j, with $j = 1,2,\ldots,l$ to perform the task i, with $i = 1,2,\ldots,n^{\text{tot}}$. Instead, the matrix $\boldsymbol{C}_{\text{p}}$ of dimension $n^{\text{tot}} \times m$ includes the completion time forced by the human supervisor. Each element $C_{\text{p}}(i,j)$ defines to whom the task i, with $i = 1,2,\ldots,n^{\text{tot}}$ is assigned to the robot j, with $j = 1,2,\ldots,l$ and when the task i must be completed.

7.3 Methodology

In this thesis, a centralized approach is adopted since the human supervisor must have the possibility to approve the final plan and to take action (e.g. change capabilities for a given task or assign a task to a particular robot) about the plan in two different situations: when the plan is in execution; and when the human supervisor does not approve the plan.

Once the plan is approved, assigned tasks are executed in a completely autonomous fashion. That is, each robot can move toward the assigned position and autonomously execute its task, counting only on its capabilities. We also assume that, once scheduled, each robot is able to perform the planned tasks successfully.

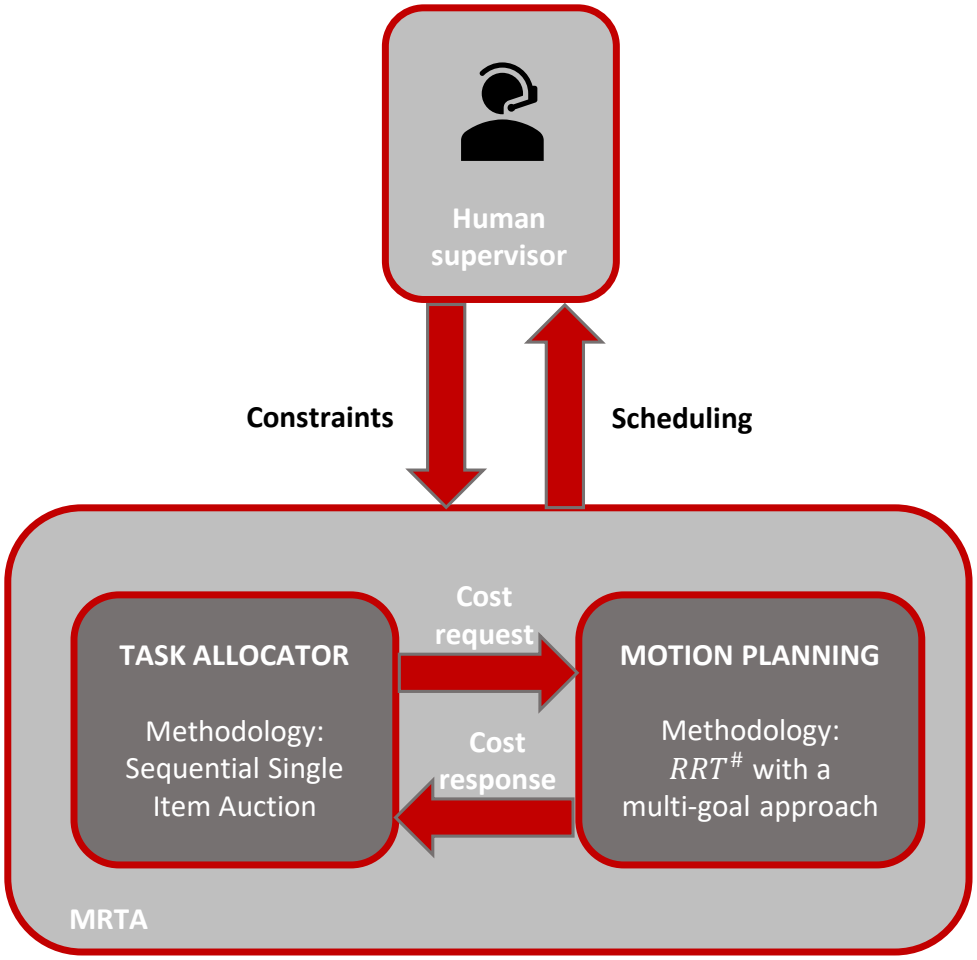


Fig. 7.3 Overview of the methodology for each block of the HMRTA.

The centralized system is composed of a task allocator based on a sequential single-item auction, and a centralized motion planner based on RRT$^{\#}$ with a multi-goal approach. These blocks continuously interact to compute all the paths connecting robots and tasks and estimate their costs in order to compute the plan which will be checked by the human supervisor, as shown in Figure 7.3.

The communication between the two blocks is assumed as *ideal* –without delays and losses of information.

In the following section, each algorithm is detailed.

7.3.1 Algorithms

Auction-based Task Allocator

A traditional auction is composed of two steps: the *bidding step* and the *winner determination step*. In the *bidding step*, the auctioneer informs the robots about the tasks for sale. Then, each robot evaluates the tasks, calculates the bid, and returns the bid to the auctioneer. Then, during the *winner determination step*, the auctioneer determines the winner for each task and informs the winning robots. These two steps compose the so-called *round* of the auction.

In our problem, we choose a sequential single-item (SSI) auction where the auctioneer (the centralized task allocator) sells one item (task) for each round in an order selected respecting the priority of targets s.

In the proposed strategy, a centralized motion planner is called at each round to compute the bids of all robots to perform a task. In fact, the bid returned by a robot includes the cost of moving toward the task's position.

During the *winner determination step*, the auctioneer assigns the task to the robot r_{best} with the right capabilities and with the lowest bidder.

In particular, our algorithm based on a sequential single-item auction with the decision of the human supervisor is summarized in Algorithm 9.

The **inputs** of the task allocator are: the set of all the tasks $\boldsymbol{K}^{\text{tot}}$, the set of robots $\boldsymbol{R}$, the possible instant of perturbation t_{new}, the set of priority for each sub-operation $\boldsymbol{GP}$, the matrix with the combination of robots and capabilities $\boldsymbol{RC}(\boldsymbol{t})$, the matrix with the combination of tasks and capabilities $\boldsymbol{TC}$, the operational space $\boldsymbol{X}$ including obstacles and free space, the vector of robots' positions $\boldsymbol{X}_{\text{r}}$, the vector of tasks' positions $\boldsymbol{X}_{\text{k}}$, the matrix with the assignment of tasks to robots made by the human supervisor $\boldsymbol{C}_{\text{p}}$, and the matrix $\boldsymbol{TC}^{\text{H}}$ with the assignment of sensors to tasks made by the human supervisor.

The Algorithm 9 is split into two macro steps: **Initialization** and **Auction**.

The **Initialization** is fundamental in order to create and initialize variables essential for the auction.

$\boldsymbol{C}$ is the matrix of the completion time for all the tasks, where the element $C(k,r)$ denotes the completion time of the task $k \in \boldsymbol{K}^{\text{tot}}$ performed by the robot $r \in \boldsymbol{R}$ (line

Algorithm 9: Task allocation algorithm based on Auction

Input: $\boldsymbol{K}^{\text{tot}}, \boldsymbol{R}, t_{\text{new}}, \boldsymbol{GP}, \boldsymbol{RC}(t), \boldsymbol{TC}, \boldsymbol{X}, \boldsymbol{X}_{\text{obs}}, \boldsymbol{X}_{\text{r}}, \boldsymbol{X}_{\text{k}}, \boldsymbol{C}_{\text{p}}, \boldsymbol{TC}^{\text{H}}$
Initialization:
$\boldsymbol{C} \leftarrow zeros(n^{\text{tot}}, m)$
$\boldsymbol{T}_0 \leftarrow zeros(n^{\text{tot}})$
if $\boldsymbol{TC}^{\text{H}}$ *is not empty* **then**
    $\boldsymbol{TC} \leftarrow \text{HumanChoiceCapabilities}(\boldsymbol{TC}^{\text{H}})$
Auction:
$\boldsymbol{M}_{\text{st}} \leftarrow \text{CreationStaticMask}(\boldsymbol{TC}, \boldsymbol{RC})$
if $\text{ControllingFeasibleOperation}(\boldsymbol{M}_{\text{st}}) = \text{True}$ **then**
    $\boldsymbol{L}_{\text{pr}} \leftarrow \text{CreationListPrioritizedTask}(\boldsymbol{K}^{\text{tot}}, \boldsymbol{GP})$
    $(\boldsymbol{K}_{\text{forced}}, \boldsymbol{C}) \leftarrow \text{HumanChoiceTasksRobot}(\boldsymbol{C}_{\text{p}}, \boldsymbol{L}_{\text{pr}})$
    foreach *task* $k_{\text{pr}} \in \boldsymbol{L}_{\text{pr}}$ **do**
        if $k_{\text{pr}} \notin \boldsymbol{K}_{\text{forced}}$ **then**
            $(\boldsymbol{M}_{\text{dyn}}, t_0^{\text{exp}}(k_{\text{pr}})) \leftarrow$
            $\text{CreationDynamicMask}(\boldsymbol{M}_{\text{st}}, \boldsymbol{C}, \boldsymbol{K}^{\text{tot}}, k_{\text{pr}}, t_{\text{new}})$
            $(\textbf{Costs}, \textbf{Times}) \leftarrow \text{GetCosts}(\boldsymbol{M}_{\text{st}}, \boldsymbol{X}, \boldsymbol{X}_{\text{r}}, \boldsymbol{x}(k_{\text{pr}}))$
            if $\text{ControllingAvailabilityRobots}(k_{\text{pr}}, \boldsymbol{M}_{\text{dyn}}) = \text{False}$ **then**
                $r_{\text{best}} \leftarrow \text{SelectionBestRobot}(\textbf{Costs}, \boldsymbol{M}_{\text{st}})$
                $t_0(k_{\text{pr}}) \leftarrow \max(\boldsymbol{C}(\text{k}, \text{r}_{\text{best}})) \quad \forall k \in \boldsymbol{K}^{\text{tot}}$
            else
                $r_{\text{best}} \leftarrow \text{SelectionBestRobot}(\textbf{Costs}, \boldsymbol{M}_{\text{dyn}})$
                $t_0(k_{\text{pr}}) \leftarrow t_0^{\text{exp}}(k_{\text{pr}})$
            $C(k_{\text{pr}}, r_{\text{best}}) \leftarrow t_0(k_{\text{pr}}) + Times(r_{\text{best}})$
            $\boldsymbol{x}(r_{\text{best}}) \leftarrow \boldsymbol{x}(k_{\text{pr}})$
else
    return warning to supervisor

3). T_0 is the vector of the starting times of all the tasks n^{tot}, where $t_0(k)$ is the starting time of task $k \in \boldsymbol{K}^{\text{tot}}$ (line 4).

$\boldsymbol{TC}^{\text{H}}$ is the matrix with the capabilities assigned to the tasks by the human supervisor. Each element $TC^{\text{H}}_{i,j}$ defines if the task i, with $i = 1,2,...,n^{\text{tot}}$ requires the capability j, with $j = 1,2,...,l$. If the $\boldsymbol{TC}^{\text{H}}$ matrix is empty, the human supervisor has not added any constraint on the capabilities for the tasks. Otherwise, the function HumanChoiceCapabilities updates the $\boldsymbol{TC}$ matrix with the information of $\boldsymbol{TC}^{\text{H}}$ (line 6).

The **Auction** represents the main task allocation algorithm. In this macro step, if at least a robot performing each task exists, the auction handles sequentially each task according to the list of prioritized tasks.

In the following, we describe each function of Algorithm 9:

- CreationStaticMask: the main goal of this function is to create a static mask that defines which robot is able to do which task. In particular, the $\boldsymbol{M}_{\text{st}}$ is a boolean matrix, where the element $M_{\text{st}}(k,r)$ denotes if the robot r is able to perform the task k;
- ControllingFeasibleOperation: given the static mask $\boldsymbol{M}_{\text{st}}$, this function controls if at least a robot is able to perform each task. If not, the task allocation cannot solve the problem and the function returns a False state, warning the supervisor (line 25). Otherwise, the auction can be performed;
- CreationListPrioritizedTask: this function computes the list $\boldsymbol{L}_{\text{pr}}$, in which each task is ordered sequentially starting with the one with the highest priority. If two tasks have the same priority, then the algorithm randomly chooses the task to be evaluated first. This situation could happen when there are sub-operations with the same priority;
- HumanChoiceTasksRobot: given the matrix with the assignment of tasks to robots $\boldsymbol{C}_{\text{p}}$ forced by the human supervisor and the list of the prioritized tasks $\boldsymbol{L}_{\text{pr}}$, this function updates the matrix of the completion time $\boldsymbol{C}$ and computes the vector of the tasks already assigned by the human supervisor $\boldsymbol{K}_{\text{forced}}$;
- CreationDynamicMask: if the selected task k_{pr} is not located in $\boldsymbol{K}_{\text{forced}}$ (line 13), the task k_{pr} has not already been allocated and, then, the auction tries to assign

the task. Given the static mask $\boldsymbol{M}_{\text{st}}$, the matrix of the completion time $\boldsymbol{C}$, the lists of sub-operations with the corresponding sequences between tasks $\boldsymbol{K}_{\text{tot}}$, the task to be handled k_{pr}, and the eventual instant of perturbation t_{new} (if we are in the re-allocation phase), this function computes the time in which the task k_{pr} should start $t_0^{\text{exp}}(k_{\text{pr}}) \in \boldsymbol{T}_{\boldsymbol{0}}^{\text{exp}}$ and the dynamic mask $\boldsymbol{M}_{\text{dyn}}$. $\boldsymbol{M}_{\text{dyn}}$ is a boolean matrix that allows the algorithm to know which robot is busy when the algorithm is assigning the task k_{pr} ($t_0^{\text{exp}}(k_{\text{pr}})$) and does not have the capabilities to perform the task k_{pr}. For completeness, the dimensions of the dynamic mask $\boldsymbol{M}_{\text{dyn}}$ are the same as the static mask $\boldsymbol{M}_{\text{st}}$;

- GetCosts: this function provides the interaction with the motion planner implementing the *bidding* step of the auction. Given the static mask $\boldsymbol{M}_{\text{st}}$, the operational space $\boldsymbol{X}$, the robots' positions $\boldsymbol{X}_{\text{r}}$, the task position $\boldsymbol{x}(k_{\text{pr}})$, the motion planning computes the costs (**Costs**) and execution times (**Times**) to reach the task $\boldsymbol{x}(k_{\text{pr}})$ by each robot that has the capabilities. In our problem, we solely consider the time required to reach the position of a task, as we assume that the execution time of the task is typically negligible than the time to reach its position.

  More details about this function have been provided below with the description of Algorithm 10;

- ControllingAvailabilityRobots: this function controls if at least one robot is available to perform the task k_{pr} at the expected starting time $t_0^{\text{exp}}(k_{\text{pr}})$ exists by checking the dynamic mask $\boldsymbol{M}_{\text{dyn}}$. If it does not exist, the function returns a False value. This condition means that at the instant of assignment ($t_0^{\text{exp}}(k_{\text{pr}})$) there is no free robot because robots that would have the capabilities to perform the task k_{pr} are busy;

- SelectionBestRobot: this function provides the second step of the auction: *the winner determination step*.

  If no robot can perform the task k_{pr} at the expected starting time $t_0^{\text{exp}}(k_{pr})$ (i.e. ControllingAvailabilityRobots() = False), the SelectionBestRobot function selects the robot with the minimum cost to perform the task k_{pr} but considering the static mask $\boldsymbol{M}_{\text{st}}$. This detail is important because in this case, the choice of the best robot (r_{best}) is made only in consideration of who has the capabilities to do it and thus not considering the availability at the expected starting time

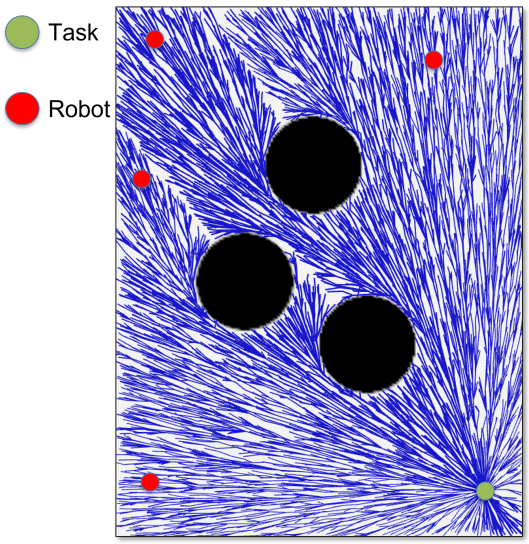


Fig. 7.4 Example of the exploration graph constructed by the RRT$^{\#}$ algorithm rooted from the task position. The graph (in blue) explores the map reaching all the robots (in red) avoiding the obstacles (in black). The computed path per each robot is the branch connecting task and robot positions.

$t_0^{\exp}(k_{\text{pr}})$. For this reason, the real starting time $t_0(k_{\text{pr}})$ for the task k_{pr} is updated considering the maximum value between the completion time of the tasks already assigned to the winner robot (line 18).

On the other hand, if at least a robot to perform the task k_{pr} exists (i.e. ControllingAvailabilityRobots() = True), the SelectionBestRobot() function selects the robot with the minimum cost to perform the task k_{pr} but, unlike the previous case, considering the dynamic mask $\boldsymbol{M}_{\text{dyn}}$ since we want to allocate the task at $t_0^{\exp}(k_{\text{pr}})$. Then, in line 21 the actual starting time ($t_0(k_{\text{pr}})$) for the task k_{pr} is updated with the expected one ($t_0^{\exp}(k_{\text{pr}})$).

Finally, the completion time for task k_{pr} is computed (line 22), and the position of the winner robot is updated with the position of task k_{pr} (line 23).

Motion planner

As previously defined, the motion planner algorithm is called several times by the task allocation algorithm with the function GetCosts. The motion planner is implemented using the RRT$^{\#}$ algorithm extended with a multi-goal strategy. In fact, in this work, the well-known RRT$^{\#}$ is exploited to construct an asymptotically

optimal graph exploring the entire map (i.e. the search space). The graph is rooted from the task position and is constructed by randomly sampling and connecting states of the search space as in [166]. Hence, we use the constructed graph to compute all the paths connecting the task position with the robot positions. In fact, as with all the RRT-based algorithms, only one branch of the graph exists connecting the origin of the graph (i.e. the task position) and any other state of the graph. This strategy is perfectly suited to the centralized task allocator because only one exploration graph is constructed to compute all the paths and their costs, instead of computing all the paths sequentially as commonly performed in the literature. An example of this strategy is shown in Figure 7.4.

Algorithm 10: The GetCosts function implementing the multi-goal RRT$^{\#}$.

```
1  GetCosts(M_st, X, X_r, x(k_pr))
2      x_0 = x(k_pr);
3      G ← {x_0};
4      for i = 0 to N do
5          x_rand ← Sample();
6          𝒢 ← Extend(𝒢, x_rand);
7          Replan(𝒢);
8      foreach x_r ∈ X_r do
9          if M_st(r) = True then
10             𝒯 ← SpanningTree(𝒢, x_r);
11             if 𝒯 = ∅ then
12                 Costs ← {NaN};
13                 Times ← {NaN};
14             else
15                 Costs ← {c(𝒯)};
16                 Times ← {t(𝒯)};
17         else
18             Costs ← {NaN};
19             Times ← {NaN};
20     return Costs, Times
```

The pseudocode of the motion planner is described in Algorithm 10. The inputs of the function are: the set $\boldsymbol{X}_{\mathrm{r}}$ with the robot positions; the position $\boldsymbol{x}(k_{\mathrm{pr}})$ of the task k_{pr}; the matrix $\boldsymbol{M}_{\mathrm{st}}$ that defines which robots have the capabilities to execute the

task k_{pr}; and the operational space $\boldsymbol{X}$ that determines the search space of the motion planning problem including obstacles.

First, the task position is added to the graph $\mathscr{G}$ as the initial state (lines 2 and 3). Then, the iterative procedure that constructs the exploration graph starts and continues until a certain number of states are added to the graph (lines 4 to 7). At each iteration, a new state $\boldsymbol{x}_{\text{rand}}$ is randomly sampled in the search space (line 5), and it is added to the graph $\mathscr{G}$ with the Extend() procedure (line 6). The Extend() procedure is an essential step of the RRT$^{\#}$ algorithm because it extends the current graph by connecting $\boldsymbol{x}_{\text{rand}}$ to the state with the minimum cost. Then, the Replan() procedure propagates all the updated costs on the graph, in order to update the graph accordingly (line 7). The Extend() and Replan() procedures are implemented exactly as in the original RRT$^{\#}$, for more details refer to [166]. After the graph is constructed, the algorithm defines the path for each robot position (lines 8 to 19). First, the algorithm verifies if the robot r with position $\boldsymbol{x}_{\text{r}} \in \boldsymbol{X}_{\text{r}}$ has the capabilities to perform the task k_{pr}. In case, the branch $\mathscr{T}$ connecting the task and robot positions is extracted from the graph (line 10), and the corresponding cost and time are included in the vector of **Costs** and **Times**, respectively. If a solution connecting the robot position $\boldsymbol{x}_{\text{r}}$ and the task position $\boldsymbol{x}(k_{\text{pr}})$ does not exist, the cost and the time related to the robot-task combination are defined as NaN (Not a Number) (lines 12 and 13). A similar condition occurs if the robot is not suitable to perform the task (lines 18 and 19). Otherwise, when a solution connecting the robot position and the task position exists, the cost is defined considering the cost function used to compute the path, i.e. the path length in this thesis. Instead, the time to reach the task position is estimated assuming that a robot moves at a constant speed. Then, the vectors **Costs** and **Times** are returned to the task allocation (line 20).

7.4 Results

In this section, the proposed task allocation and motion planning strategy is tested through simulations. The proposed strategy is implemented using the ROS (Robot Operating System) framework [181]. Specifically, the auction-based task allocation is implemented as a ROS node using Python, while the motion planner node is implemented using C++ and exploiting the OMPL (Open Motion Planning Li-

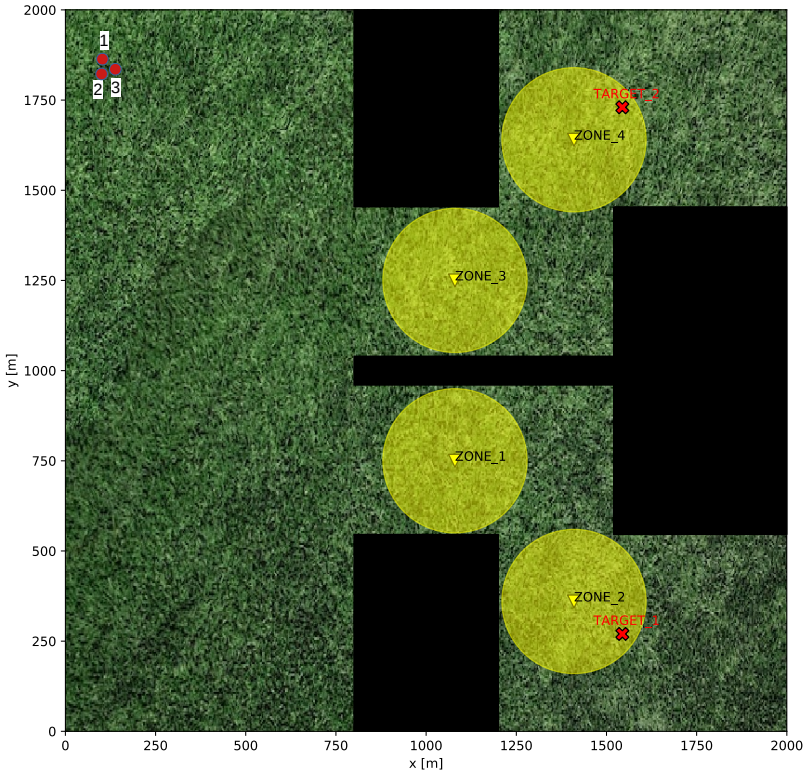


Fig. 7.5 The basic scenario evaluated in this work with the multi-robot system composed of 3 robots indicated with red circles in the upper left corner.

brary), an open-source library that contains several sampling-based motion planning algorithms [182].

In the following, the results have been split into four parts: first, we show how the proposed strategy is able to handle a basic scenario; second, the results related to dynamic scenarios with a human supervisor action are shown; third, we focus on the motion planner, showing the advantages of the proposed multi-goal strategy; finally, we show the advantages of adopting a synergetic combination of the auction-based allocation and the multi-goal RRT$^{\#}$ motion planning.

7.4.1 Basic scenario

In this paragraph, we introduce the basic scenario considering a rescue operation as shown in Figure 7.5. The main goal of the operation is to rescue two people with

Sub-operation	Tasks
1	Fix(Zone_1) → Operate(Zone_1) → Fix(Zone_2) → Operate(Zone_2) → Rescue(Target_1)
2	Fix(Zone_3) → Operate(Zone_3) → Fix(Zone_4) → Operate(Zone_4) → Rescue(Target_2)

Table 7.1 Tasks with precedence for each sub-operation considering the basic scenario of Figure 7.5.

the same priority in the areas denoted as Target_1 and Target_2, therefore, in this example, the targets (s) are 2.

The black zones are obstacles (X_{obs}), while yellow areas (Zone_1, Zone_2, Zone_3, and Zone_4) are zones to be adjusted to unlock the passage (Fix task), and, then, to be managed for example by extinguishing the fire (Operate task) to enable the navigation in that area by the robot in charge of rescue people (Rescue task).

In this example, the hierarchical structure shown in Figure 7.1 is observed. Indeed, the final goal of the operation is to rescue two targets, i.e. two people with the same priority. Thus, the sub-operations are two and are composed of tasks with precedence. The tasks for each sub-operation are described in Table 7.1.

The first sub-operation, in Table 7.1, is to handle the Zone_1, Zone_2 and Target_1 sequentially. Instead, the second sub-operation is to handle the Zone_3, Zone_4 and Target_2 sequentially. Both the sub-operations have the same priorities and, then, can be performed simultaneously. Practically, the sub-operations force that Zone_1 and Zone_3 must be adjusted and managed before Zone_2 and Zone_4 and, lastly, people can be rescued in Target_1 and Target_2.

We assumed that each robot can have at most 4 capabilities (i.e. $l = 4$). Table 7.2 shows the capabilities of each robot (***RC***) belonging to the heterogeneous multi-robot system during the entire simulation time of the basic scenario. Thus, in this case, the capabilities of each robot remain unchanged throughout the simulation.

The capabilities p_1, p_2, p_3 and p_4 are particular features that empower a robot to accomplish a particular operation. In a practical rescue scenario, these capabilities aim to enhance the robot's effectiveness in saving lives and providing assistance during the emergency situations. For example, p_1 may refer to the ability to manipulate objects within the scenario, enabling to move obstacles and clearing the path required to reach the person in need of rescue. In our simulations, this capability is used in the "fix" task. Furthermore, p_2 may improve the robot performance during nighttime rescue operations. By incorporating special equipment to rescue the person (e.g.

	p_1	p_2	p_3	p_4
r_1	x	x	x	x
r_2	x			x
r_3	x	x		x

Table 7.2 Capabilities for each robot (***RC***) considering the entire simulation time of the basic scenario.

rescue ropes) and night vision, the robot is equipped to navigate and rescue people even in low-light conditions. On the other hand, p_3 may focus on daytime rescue operations. This capability provides the robot with equipment to rescue the person but does not include night vision, limiting its effectiveness to daylight hours. Lastly, p_4 may address the specific hazard of fires encountered during rescue operations. This capability, used in the "operate" task in our simulations, equips the robot with fire extinguishers.

Table 7.3 summarizes the capabilities needed for the execution of each task.

The auction-based task allocation, through the ongoing support of the motion planner, is able to successfully manage the basic scenario. Figure 7.6 shows the resulting plan that respects the precedence constraints between tasks, the heterogeneity of the team, and the prioritization between sub-operations. The time to reach the position for each task is estimated by the motion planner, considering the robot moving at constant speed.

7.4.2 Dynamic scenario

The results of this section are obtained by evaluating the basic scenario of Figure 7.5 but considering different perturbations at different instants. Thus, the auction-based task allocation is tested by simulating a dynamic scenario, and performing a re-allocation of the basic plan.

Specifically, results show how the system is able to handle both a sensor or a robot failure, and both the intervention of the human supervisor that decides to assign a task to a specific robot.

Task	p_1	p_2	p_3	p_4
Fix zone 1	x			
Operate zone 1				x
Fix zone 2	x			
Operate zone 2				x
Fix zone 3	x			
Operate zone 3				x
Fix zone 4	x			
Operate zone 4				x
Rescue target 1			x	
Rescue target 2		x		

Table 7.3 Capabilities for each task (***TC***).

	p_1	p_2	p_3	p_4
r_1	x	x	x	x
r_2	x			
r_3	x	x		x

Table 7.4 Capabilities for each robot after a sensor failure

Figures 7.7 and Figure 7.8 show the resulting plan after two different perturbations.

In the first condition, starting from the basic scenario, the re-allocation phase is triggered at the time instant of 500 s due to a failure of capability 4 on the second robot (see Table 7.4). Thus, the task allocator is called and the plan is re-allocated (see Figure 7.7), thanks to the auction and the multi-goal motion planner.

In the second condition, starting from the basic scenario, the re-allocation phase is triggered at the time instant of 1500 s due to a total failure of the third robot. Thus, the whole system re-allocates the plan and the result is summarized in Figure 7.8.

Another simulation is performed including an action of the human supervisor.

Starting from the basic scenario, Figure 7.9 shows the plan after the action of the supervisor that forces the assignment of the task Fix(Zone_1) to the third robot.

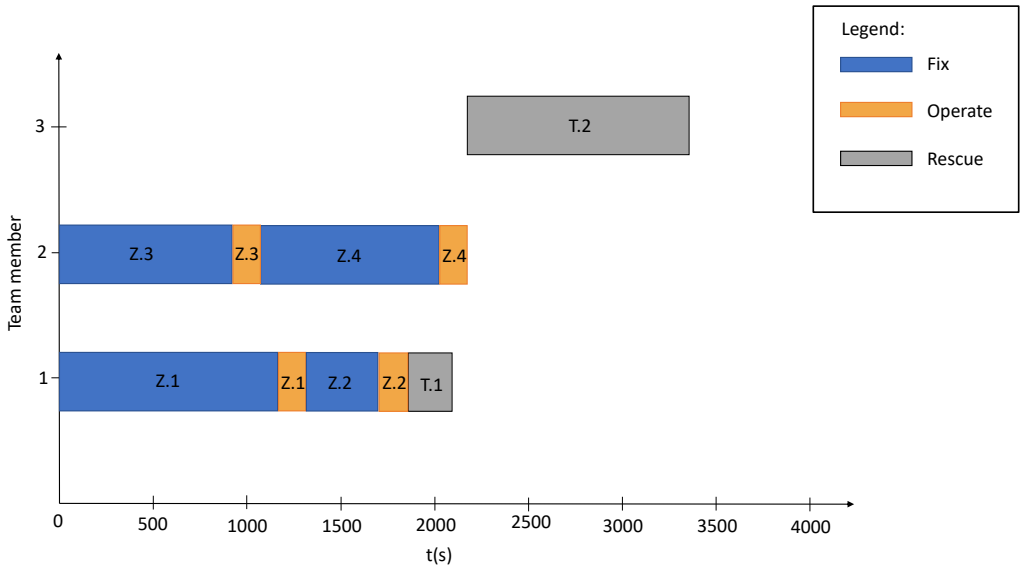


Fig. 7.6 Allocation of the basic scenario.

Here, the task allocator complies with this additional condition and allocates all other tasks accordingly, while respecting all the constraints we have detailed above.

7.4.3 Multi-goal Motion Planner

As previously defined, the motion planner plays a crucial role in the proposed strategy. Table 7.5 shows how the proposed motion planning improves the performance in terms of computational time without compromising the quality of the solution (i.e. the path length). The results of Table 7.5 compare the use of the standard $RRT^{\#}$ algorithm with the one with the multi-goal strategy proposed in this thesis. Specifically, the values of Table 7.5 are the average ones of 20 executions of the scenario of Figure 7.6.

The use of the standard RRT$^{\#}$ requires the computation of each path between a robot and task position. Hence, the motion planner is called several times in the scenario of Figure 7.6. On the contrary, the use of the multi-goal RRT$^{\#}$ reduces the number of calls of the motion planner, since it computes simultaneously the paths between a task position and all the robot positions. As a consequence, the computational time is reduced. Moreover, Table 7.5 affirms that the quality of the solution in terms of path length does not change. The solution costs of the multi-

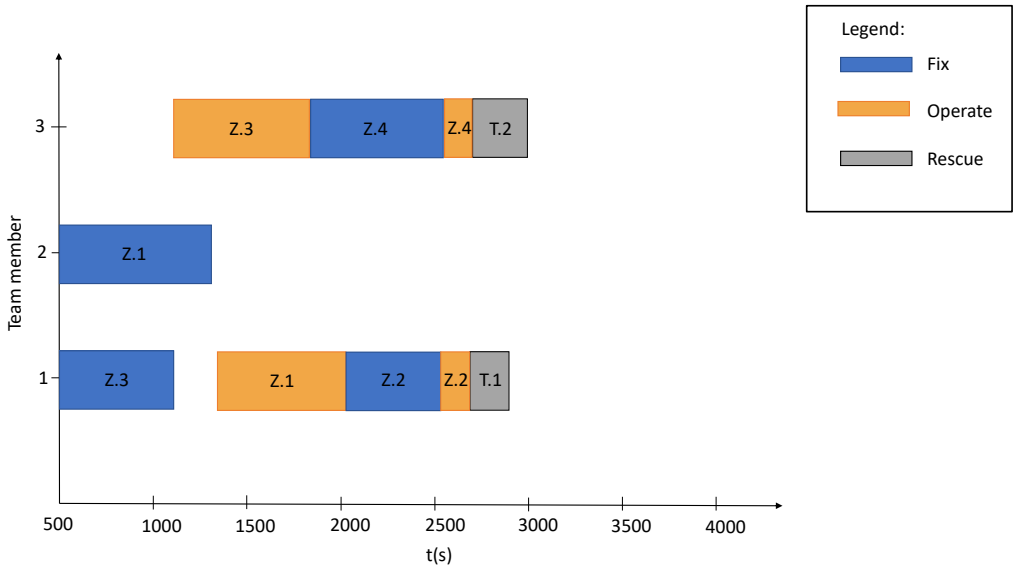


Fig. 7.7 Re-allocation starting from the basic scenario due to a failure of capability 4 of the second robot.

goal RRT# and original RRT# are very similar. The small difference is due to the non-deterministic nature of the algorithm that never computes the same solution at each execution.

Another analysis is shown in Figure 7.10, where the computational time between the multi-goal RRT# and the original RRT# is plotted as a function of the number of robots. Here, the path is computed between a fixed task position and several robots distributed in the scenario. Both multi-goal and original RRT# generate an exploration graph of 5000 states to compute the path. As result, the computational time required by the multi-goal RRT# increases slowly compared with the computational time of the original RRT#. This graph affirms that the effectiveness of the proposed multi-goal RRT# increases with the number of robots in the scenario.

7.4.4 Auction and Multi-goal Motion Planner

To demonstrate the effective synergy of the sequential-single item auction with the multi-goal motion planner, we conducted a computational time analysis for the basic scenario shown in Figure 7.5. The analysis compares the computational time required by the sequential-single item auction implementing the original RRT# with the one

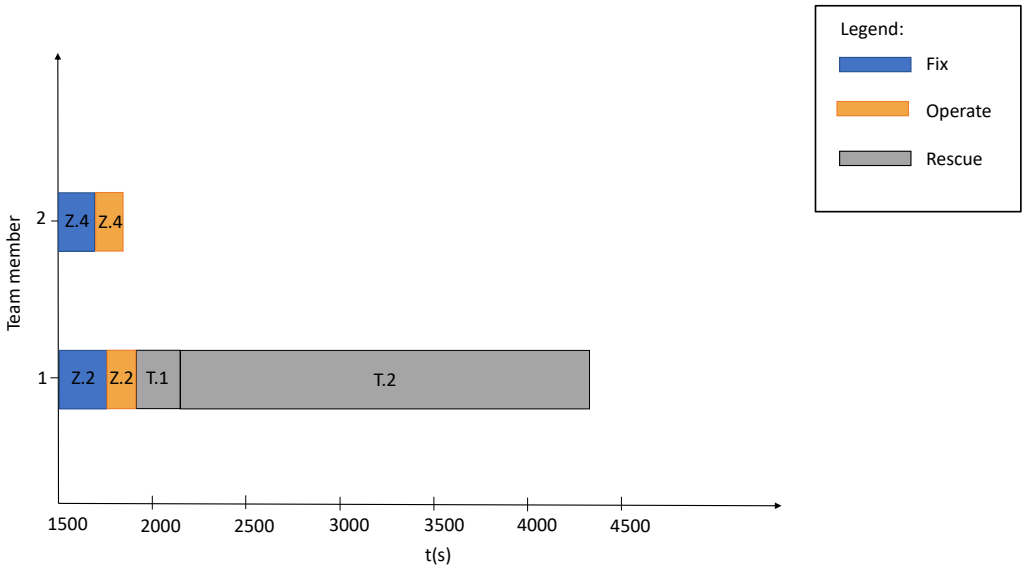


Fig. 7.8 Re-allocation due to a total failure of robot 3 starting from the basic scenario.

implementing the multi-goal RRT$^{\#}$, evaluating both the computation of the initial scheduling of Figure 7.6 and the dynamic scheduling of Figure 7.8. Simulations were executed on a laptop with the Intel Core i5-10210U processor.

Regarding the initial scheduling, the proposed approach with the multi-goal RRT$^{\#}$ computes the solution of Figure 7.6 in 2.27 seconds. Instead, the computational time required to compute a solution using the original RRT$^{\#}$ increases to 5.32 seconds. As previously discussed, this difference in the computational time is caused by the fact that the standard RRT$^{\#}$ is executed m (number of robots) times per each round of the auction. On the other hand, the multi-goal RRT$^{\#}$ is called only once per each round of the auction.

A similar trend is shown evaluating the dynamic scheduling of Figure 7.8. The use of the multi-goal planner requires 1.16 seconds, while the use of standard RRT$^{\#}$ implies a computational time of 2.03 seconds. In this scenario, the computational time is lower because the task allocation problem involves only 2 robots and 6 tasks. This test highlights the benefits introduced by the proposed approach. Moreover, as also shown in Figure 7.10, the benefits of our approach become evident as the number of robots increases.

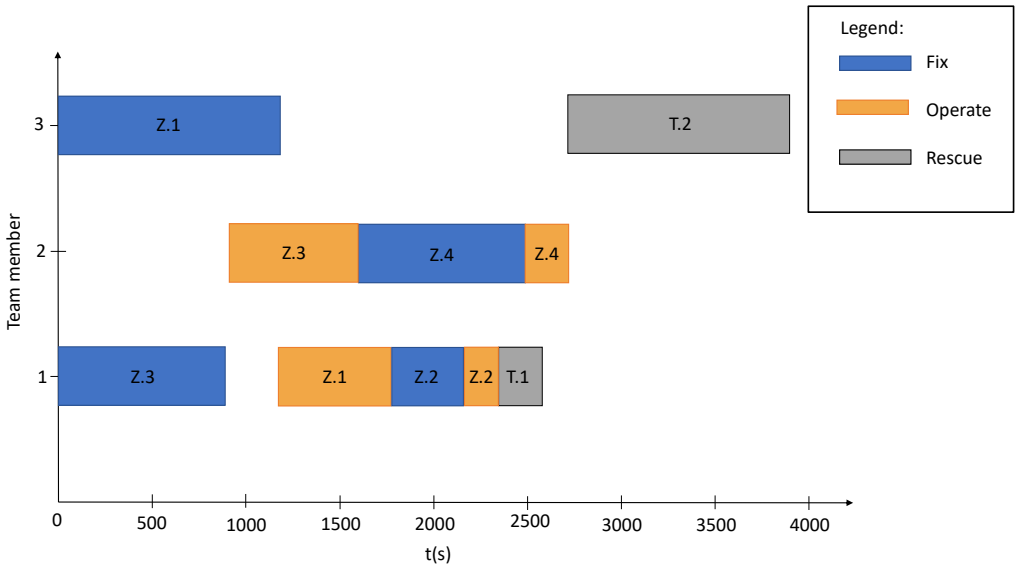


Fig. 7.9 Basic scenario with a supervisor decision. Indeed, the allocation of the task "Fix zone 1" has been assigned to the third robot by the supervisor and the remaining tasks have been allocated by the auction algorithm.

	Computational Time [s]	**Solution Cost [s]**
original RRT#	1.716 (+45%)	356.609 (-0.1%)
multi-goal RRT#	1.182	356.953

Table 7.5 Comparison between the original and multi-goal RRT# applied in the scenario of Figure 7.5.

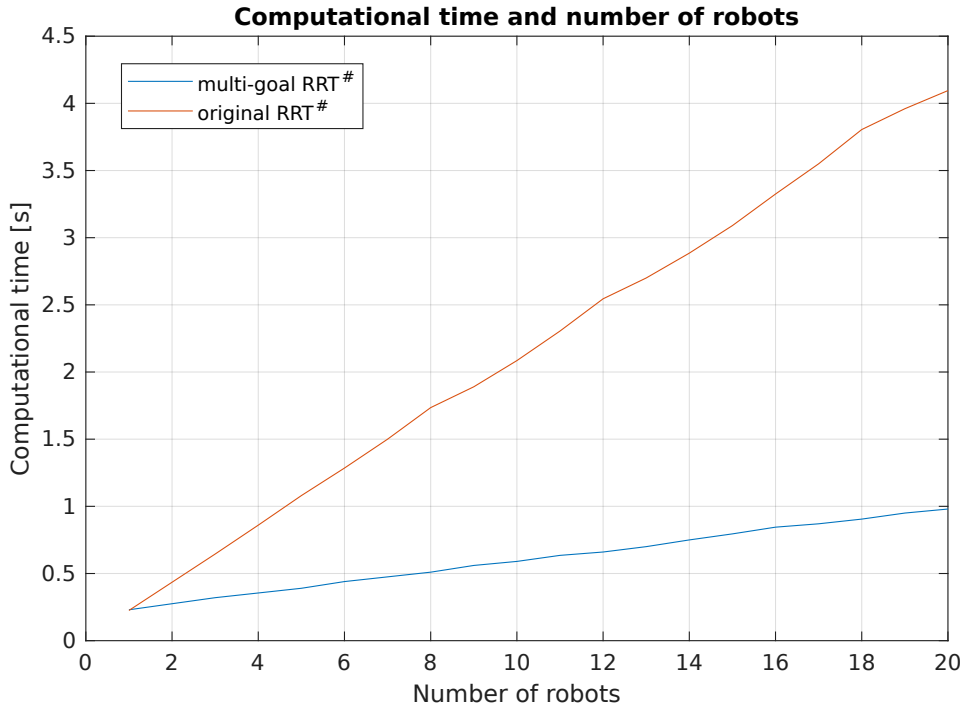


Fig. 7.10 Comparison of the computational time between the original and multi-goal RRT# as a function of the number of robots.

7.5 Discussion

In this thesis, a dynamic task allocation and a motion planning strategy for a team of heterogeneous robots are proposed by also including the interaction with a human supervisor. Specifically, the proposed solution consists of an auction-based task allocation, and a sampling-based motion planning based on the RRT# algorithm and enhanced with a multi-goal approach.

We decided to adopt a centralized architecture composed of a centralized task allocator and a centralized motion planner because it offers three important advantages First, the task allocator can directly interact with the motion planner, without avoiding delays caused by the communication with each agent of the HMRS. Second, the motion planner can parallelize path calculations. This logic could not have been adopted with the decentralized structure. Third, a centralized architecture is suitable for the interaction with a human supervisor. In this way, the supervisor has the possibility to intervene in the planning of the operation from a global point of view.

The proposed framework is tested in a simulation environment proving that our strategy is able to tackle a complex operation composed of different tasks in a dynamic scenario.

First, the proposed strategy is capable of handling the rescue operation of the basic scenario, as well as handling perturbation events, e.g. sensor and robot failures. Results indicate that our approach can handle a multi-robot heterogeneous system in a dynamic scenario respecting precedence between tasks and priorities among sub-operations having computational efficiency as the main constraint since the system must be able to re-allocate on-the-go. This peculiarity is fulfilled by two features of our methodology: *(i)* the use of an auction-based task allocation with a human supervisor that is computationally efficient compared with MILP [168] and heuristic [179] approaches; *(ii)* the adoption of a motion planner with the multi-goal approach to take full advantage of the features of the sequential single-item auction.

The proposed multi-goal motion planner introduces several benefits to the overall system. A comparative analysis conducted in this study highlights the effectiveness of the proposed multi-goal motion planner in terms of computational time compared with the single-goal motion planner. This analysis proves the advantages introduced by the proposed method in terms of the scalability of the number of robots in the system and demonstrates the superiority of the sequential-single item auction when paired with the multi-goal RRT$^{\#}$.

Furthermore, unlike [176, 177], in this study we demonstrate that the multi-goal RRT$^{\#}$ guarantees an optimal path in the exploration graph. This is an essential feature because the quality of the solution of the auction-based task allocation strictly depends on the quality of the computed paths.

Moreover, the simulations with the interaction of the human supervisor led to promising results. The human supervisor is capable of constraining the plan by forcing the assignment to a specific robot or changing the capabilities required for tasks. Also in this scenario, the auction computes a valid plan respecting the constraints of the human supervisor. This is an important achievement since in the auction literature the human supervisor is rarely included [39, 169, 43].

Despite the promising results, the proposed approach is not exempt from limitations. First, we do not account for the *stochasticity* of the duration of a task due to an event that was not predicted in the scenario. Second, although in the interest of the system's responsiveness, the solution obtained by the task allocation is *suboptimal,*

since the adopted task allocation is based on a single-item auction. Our approach does not also take into account the *possibility of collaborations* between robots to perform tasks, nor does it contemplate temporal windows or deadlines for task completion. Moreover, the formulation of a low-level controller is required for the practical execution of the task on hardware.

The analysis of the limitations paves the way for possible improvements in the proposed strategy. For example, the *uncertainty* can be included in the estimation of the task execution time through the consideration of a specific probability distribution, along the lines of [183].

The suboptimality of the solution can be improved without affecting too much the computational efficiency by complementing the auction with heuristic approaches [184].

Collaborative tasks may be contemplated, similar to [39], where more than one agent collaborates in executing a task, together with time constraints in the execution times. The inclusion of all these aspects will affect the formulation and efficiency of the optimization problem –an aspect that notoriously leads to significant trade-offs.

In addition, a future implementation on hardware will call for the design of a low-level controller to materially execute the planned task once scheduled. Several well-established techniques have been proposed in the literature, such as in [185], where the authors present a control methodology for a mobile robot in dynamic environments that contain both fixed and moving unforeseeable obstacles.

Furthermore, different operational aspects can involve different criteria for the design of the objective function, such as the minimization of the risk of the operation [186, 187], the travel distance, or the fuel consumption. The selection of these criteria could be operated automatically, or by the human operator according to his analysis of the operational scenario.

Finally, also unreliable communications between the robots and the task allocation unit should be considered and managed. This is a crucial issue in critical scenarios, such as in rescue operations in adverse weather conditions [169].

Part III

Part 3 Conclusions

Chapter 8

Conclusions

This thesis investigates the interaction between humans and robots, aiming to develop innovative technologies that enhance human lives through an *interdisciplinary* approach. What makes this approach so essential?

The introduction of robots into society was initially intended to alleviate physical labour, leaving the more conceptual demanding tasks to humans. However, the pervasiveness of robotics and artificial intelligence (AI) has surpassed this original vision. We are moving towards a society in which these technologies not only improve but also invade the conceptual work.

The rapid advancement of robotics and AI has raised profound questions about our place in the future. What will distinguish us from these intelligent machines? Our ability to approach problems in an *interdisciplinary* way, considering both immediate consequences and the broader human impact, is a unique strength that robots and AI may struggle to replicate. This thesis uses this *interdisciplinary* approach, advocating for research that prioritizes human needs and well-being.

This methodology is applied to address the socially-aware navigation challenge, aiming to devise two game theory-based algorithms that prioritize not only the physical safety but also the psychological well-being of pedestrians. The performance of the algorithms is evaluated through *quantitatively* and *qualitatively* analysis. The first approach, with an engineering focus, employs quantitative methods to evaluate the efficiency and naturalness of the generated trajectories. In contrast, the second approach delved into the psychological aspects, exploring the extent to which the

generated trajectories can be anthropomorphized by humans and therefore perceived as socially acceptable.

Moreover, the *interdisciplinary* approach is also applied to address mission planning in this thesis. While the state of the art of managing missions in critical environments mainly focuses on developing fully autonomous systems, this thesis takes a more comprehensive approach by considering both the engineering aspects and the ethical and legal implications of the decision-making process in a critical environment.

The **take-home messages** regarding this thesis are the following:

- *Bridging the gap between technology and human science.*

  The coexistence of humans and robots is a problem that technology alone will not be able to solve. It is necessary to involve the human sciences, such as psychology, to address this issue.

- *Designing for Acceptance: The Role of Anthropomorphism in Robotic Design.*

  One crucial aspect to consider in human-robot interaction is the anthropomorphization of robots. By making robots more human-like in appearance and behavior, we can increase their acceptability to humans.

- *Optimizing Human-Robot Interactions: A Game-Theoretic Approach.*

  Game theory, a mathematical framework used in economics, has shown promise in improving the level of anthropomorphization of robots. By modeling the interactions between humans and robots as a game, we can identify strategies that can lead to more cooperative and mutually beneficial outcomes.

- *Unlocking the Robotics Revolution: the power of Open-Source Tools.*

  Open-source tools democratize access to a shared pool of expertise and innovation. ROS (Robot Operating System) is one such tool that has revolutionised the field of robotics, democratising access beyond the confines of academia. ROS has accelerated technological advances in robotics by leveraging the collective knowledge of a global community of coders and academics.

- *Beyond the Motion: The Impact of Robot Appearance on Anthropomorphism.*

To evaluate the anthropomorphism of the robot's motion, the robot's appearance should not be overlooked, as it can significantly influence people's perception of its movements.

- *Laboratory prototypes lack readiness for real-world applications.*

  Research-based implementations often lack the necessary maturity for real applications because they focus on validating theoretical concepts in controlled laboratory environments. The simulated experimentation phase often encounters similar limitations, leading to the need for significant re-evaluation and adaptation when moving from prototypes to real-world applications.

- *Bridging the gap between technology and ethical consequences.*

  Human supervision is essential for robotic systems deployed in critical environments where final decisions have life-and-death consequences. This aspect should be considered by all researchers working in this field as the ethics associated with technologies are not completely separate from the technology itself but are an integral part of it.

Motivated by the findings presented in this thesis, I would suggest several directions for future research to tackle the challenges identified in this work.

- *Testing of the algorithms in a real environment.*

  To comprehensively evaluate the anthropomorphism of robot movements, conducting real-world tests in authentic settings (such as hospitals or office corridors) would offer more reliable insights. These tests could allow for a better understanding of how human participants interact with and perceive the robot. Moreover, researchers could investigate whether interactions between humans and robot follow a peer-to-peer model or a leader-follower logic.

- *Multiscale Integration for Enhanced Performance in Crowded Scenarios.*

  The numerical simulations presented in Chapter 4 suggest that the performance of the proposed GTSFM decreases as the environment becomes highly crowded. In this regard, we envisage integrating multiscale approaches in our model [122] toward generating specialized behaviors according to the spatial scale of interpersonal interactions.

- *Realistic Mission Planning: adding new features.*

  To enhance the realism and effectiveness of mission planning, key features need to be incorporated into the system. These features should address the challenges of stochasticity of task duration, collaborative tasks, and realistic communication logic.

  Stochasticity, or unpredictable changes in task durations, can significantly impact mission execution. To effectively handle these variables, the planning system should incorporate probabilistic modeling and adaptive strategies. Collaborative task allows multiple robots to work having a shared goal, enhancing teamwork and decreasing the time to perform that particular task. Moreover, addressing the issue of unreliable communication between robots and the task allocation unit is crucial, especially in critical scenarios like rescue operations in hazardous environments.

  By incorporating these features, mission planning can become more adaptable, collaborative, and responsive to real-world situations. This will lead to more efficient and successful mission execution in complex and dynamic environments.

References

[1] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 261–268, IEEE, 2009.

[2] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, "Thör: Human-robot navigation data collection and accurate motion trajectories dataset," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 676–682, 2020.

[3] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2010.

[4] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[5] "Create3 mobile base." https://iroboteducation.github.io/create3_docs/. Accessed: 2023.

[6] "Zed2 body tracking overview | stereolabs." https://www.stereolabs.com/docs/body-tracking/. Accessed: 2023.

[7] N. Spatola, B. Kühnlenz, and G. Cheng, "Perception and evaluation in human–robot interaction: The human–robot interaction evaluation scale (hries)—a multicomponent approach of anthropomorphism," *International Journal of Social Robotics*, vol. 13, no. 7, pp. 1517–1539, 2021.

[8] S. M. Hazarika and U. S. Dixit, "Robotics: History, trends, and future directions," *Introduction to Mechanical Engineering*, pp. 213–239, 2018.

[9] S. Phuyal, D. Bista, and R. Bista, "Challenges, opportunities and future directions of smart manufacturing: a state of art review," *Sustainable Futures*, vol. 2, p. 100023, 2020.

[10] S. Sarker, L. Jamal, S. F. Ahmed, and N. Irtisam, "Robotics and artificial intelligence in healthcare during covid-19 pandemic: A systematic review," *Robotics and autonomous systems*, vol. 146, p. 103902, 2021.

[11] B. Siciliano and O. Khatib, *Robotics and the Handbook*, p. 1–6. Springer International Publishing, 2016.

[12] L. Onnasch and E. Roesler, "A taxonomy to structure and analyze human–robot interaction," *International Journal of Social Robotics*, vol. 13, no. 4, pp. 833–849, 2021.

[13] L. Battistuzzi, C. T. Recchiuto, and A. Sgorbissa, "Ethical concerns in rescue robotics: a scoping review," *Ethics and Information Technology*, vol. 23, no. 4, pp. 863–875, 2021.

[14] C. Freschi, V. Ferrari, F. Melfi, M. Ferrari, F. Mosca, and A. Cuschieri, "Technical review of the da vinci surgical telemanipulator," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 9, no. 4, pp. 396–406, 2013.

[15] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial intelligence*, vol. 114, no. 1-2, pp. 3–55, 1999.

[16] H. Asoh, S. Hayamizu, I. Hara, Y. Motomura, S. Akaho, and T. Matsui, "Socially embedded learning of the office-conversant mobile robot jijo-2," in *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence - Volume 2*, pp. 880–887, 1997.

[17] M. Takahashi, T. Suzuki, H. Shitamoto, T. Moriguchi, and K. Yoshida, "Developing a mobile robot for transport applications in the hospital domain," *Robotics and Autonomous Systems*, vol. 58, no. 7, pp. 889–899, 2010.

[18] E. Zalama, J. G. García-Bermejo, S. Marcos, S. Domínguez, R. Feliz, R. Pinillos, and J. López, "Sacarino, a service robot in a hotel environment," in *ROBOT2013: First Iberian Robotics Conference: Advances in Robotics, Vol. 2*, pp. 3–14, Springer, 2014.

[19] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, 2015.

[20] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[21] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research: The 14th International Symposium ISRR*, pp. 3–19, Springer, 2011.

[22] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005.

[23] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

[24] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.

[25] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5725–5734, 2019.

[26] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[27] Y. Gao and C.-M. Huang, "Evaluation of socially-aware robot navigation," *Frontiers in Robotics and AI*, vol. 8, p. 721317, 2022.

[28] C. I. Greenberg, M. J. Strube, and R. A. Myers, "A multitrait-multimethod investigation of interpersonal distance," *Journal of Nonverbal Behavior*, vol. 5, pp. 104–114, 1980.

[29] E. T. Hall, *The hidden dimension*, vol. 609. Garden City, NY: Doubleday, 1966.

[30] C. Bartneck, D. Kulić, E. Croft, and S. Zoghbi, "Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots," *International journal of social robotics*, vol. 1, pp. 71–81, 2009.

[31] M. Joosse, A. Sardar, M. Lohse, and V. Evers, "Behave-ii: The revised set of measures to assess users' attitudinal and behavioral responses to a social robot," *International journal of social robotics*, vol. 5, pp. 379–388, 2013.

[32] A. Waytz, J. Cacioppo, and N. Epley, "Who sees human? the stability and importance of individual differences in anthropomorphism," *Perspectives on Psychological Science*, vol. 5, no. 3, pp. 219–232, 2010.

[33] J. Fink, "Anthropomorphism and human likeness in the design of robots and human-robot interaction," in *International conference on social robotics*, pp. 199–208, Springer, 2012.

[34] S. Suárez, J. Collins, and B. López, "Improving rescue operations in disasters: approaches about task allocation and re-scheduling," in *The 24rd Annual Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG)*, pp. 66–75, 2005.

[35] R. Simmons, S. Singh, D. Hershberger, J. Ramos, and T. Smith, "First results in the coordination of heterogeneous robots for large-scale assembly," in *Experimental Robotics VII*, pp. 323–332, Springer, 2001.

[36] L. E. Parker, "Alliance: An architecture for fault tolerant multirobot cooperation," *IEEE transactions on robotics and automation*, vol. 14, no. 2, pp. 220–240, 1998.

[37] Y. Guo, L. E. Parker, and R. Madhavan, "Towards collaborative robots for infrastructure security applications," in *Proceedings of The 2004 International Symposium on Collaborative Technologies and Systems*, pp. 235–240, Citeseer, 2004.

[38] D. Albani, J. IJsselmuiden, R. Haken, and V. Trianni, "Monitoring and mapping with robot swarms for agricultural applications," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2017.

[39] C. Wang, X. Wen, Y. Niu, L. Wu, D. Yin, and J. Li, "Dynamic task allocation for heterogeneous manned-unmanned aerial vehicle teamwork," in *2018 Chinese Automation Congress (CAC)*, pp. 3345–3349, IEEE, 2018.

[40] K. Kurowski and O. v. Stryk, "Online interaction of a human supervisor with multi-robot task allocation," in *Intelligent Autonomous Systems 13*, pp. 965–978, Springer, 2016.

[41] W. Dai, H. Lu, J. Xiao, Z. Zeng, and Z. Zheng, "Multi-robot dynamic task allocation for exploration and destruction," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, pp. 455–479, 2020.

[42] E. Nunes and M. Gini, "Multi-robot auctions for allocation of tasks with temporal constraints," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.

[43] S. Moon, E. Oh, and D. H. Shim, "An integral framework of task assignment and path planning for multiple unmanned aerial vehicles in dynamic environments," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1, pp. 303–313, 2013.

[44] C. Pippin, H. Christensen, and L. Weiss, "Performance based task assignment in multi-robot patrolling," in *Proceedings of the 28th annual ACM symposium on applied computing*, pp. 70–76, 2013.

[45] C. Torras, "Service robots for citizens of the future," *European Review*, vol. 24, no. 1, pp. 17–30, 2016.

[46] R. Philippsen and R. Siegwart, "Smooth and efficient obstacle avoidance for a tour guide robot," in *Proceedings. 2003 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 446–451, 2003.

[47] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[48] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[49] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1928–1935, IEEE, 2008.

[50] H. Hewawasam, M. Y. Ibrahim, and G. K. Appuhamillage, "Past, present and future of path-planning algorithms for mobile robot navigation in dynamic environments," *IEEE Open Journal of the Industrial Electronics Society*, vol. 3, pp. 353–365, 2022.

[51] H. Kivrak, F. Cakmak, H. Kose, and S. Yavuz, "Social navigation framework for assistive robots in human inhabited unknown environments," *Engineering Science and Technology, an International Journal*, 2020.

[52] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[53] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[54] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model," *International Journal of Social Robotics*, vol. 6, no. 3, pp. 443–455, 2014.

[55] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, IEEE, 2017.

[56] J. Nash, "Non-cooperative games," *Annals of mathematics*, pp. 286–295, 1951.

[57] C. Cenedese, F. Fabiani, M. Cucuzzella, J. M. A. Scherpen, M. Cao, and S. Grammatico, "Charging plug-in electric vehicles as a mixed-integer aggregative game," in *IEEE 58th Conference on Decision and Control*, pp. 4904–4909, 2019.

[58] D. Bauso, H. Tembine, and T. Başar, "Opinion dynamics in social networks through mean-field games," *SIAM Journal on Control and Optimization*, vol. 54, pp. 3225–3257, Jan. 2016.

[59] N. Quijano *et al.*, "The role of population games and evolutionary dynamics in distributed control systems: The advantages of evolutionary game theory," *IEEE Control Systems Magazine*, vol. 37, no. 1, pp. 70–97, 2017.

[60] M. Ye, L. Zino, A. Rizzo, and M. Cao, "Game-theoretic modeling of collective decision making during epidemics," *Physical Review E*, vol. 104, no. 2, p. 024314, 2021.

[61] J. Tanimoto, A. Hagishima, and Y. Tanaka, "Study of bottleneck effect at an emergency evacuation exit using cellular automata model, mean field approximation analysis, and game theory," *Physica A: statistical mechanics and its applications*, vol. 389, no. 24, pp. 5611–5618, 2010.

[62] B. L. Mesmer and C. L. Bloebaum, "Modeling decision and game theory based pedestrian velocity vector decisions with interacting individuals," *Safety science*, vol. 87, pp. 116–130, 2016.

[63] Y. Rahmati, A. Talebpour, A. Mittal, and J. Fishelson, "Game theory-based framework for modeling human–vehicle interactions on the road," *Transportation research record*, vol. 2674, no. 9, pp. 701–713, 2020.

[64] C. Kim and J.-S. Won, "A fuzzy analytic hierarchy process and cooperative game theory combined multiple mobile robot navigation algorithm," *Sensors*, vol. 20, no. 10, p. 2827, 2020.

[65] S. M. LaValle and S. Hutchinson, "Game theory as a unifying structure for a variety of robot tasks," in *Proceedings of 8th IEEE International Symposium on Intelligent Control*, pp. 429–434, IEEE, 1993.

[66] Y. Meng, "Multi-robot searching using game-theory based approach," *International Journal of Advanced Robotic Systems*, vol. 5, no. 4, p. 44, 2008.

[67] S. Tadokoro, M. Hayashi, Y. Manabe, Y. Nakami, and T. Takamori, "On motion planning of mobile robots which coexist and cooperate with human," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 2, pp. 518–523, IEEE, 1995.

[68] F. Hoeller, D. Schulz, M. Moors, and F. E. Schneider, "Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1260–1265, IEEE, 2007.

[69] A. Turnwald, D. Althoff, D. Wollherr, and M. Buss, "Understanding human avoidance behavior: interaction-aware decision making based on game theory," *International Journal of Social Robotics*, vol. 8, no. 2, pp. 331–351, 2016.

[70] J. Cheng, H. Cheng, M. Q.-H. Meng, and H. Zhang, "Autonomous navigation by mobile robots in human environments: A survey," in *2018 IEEE international conference on robotics and biomimetics (ROBIO)*, pp. 1981–1986, IEEE, 2018.

[71] A. Antonucci *et al.*, "Socially aware robot navigation," 2022.

[72] G. Ferrer, A. G. Zulueta, F. H. Cotarelo, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous robots*, vol. 41, no. 4, pp. 775–793, 2017.

[73] P. Ratsamee, Y. Mae, K. Kamiyama, M. Horade, M. Kojima, and T. Arai, "Social interactive robot navigation based on human intention analysis from face orientation and human path prediction," *Robomech Journal*, vol. 2, no. 1, pp. 1–18, 2015.

[74] A. Turnwald, *Human-like Motion Planning in Populated Environments*. PhD thesis, Technische Universität München, 2019.

[75] N. H. Singh and K. Thongam, "Neural network-based approaches for mobile robot navigation in static and moving obstacles environments," *Intelligent Service Robotics*, vol. 12, no. 1, pp. 55–67, 2019.

[76] Z. Xie, P. Xin, and P. Dames, "Towards safe navigation through crowded dynamic environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4934–4940, IEEE, 2021.

[77] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 285–292, IEEE, 2017.

[78] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 international conference on robotics and automation (ICRA)*, pp. 6015–6022, IEEE, 2019.

[79] Y. Yildirim and E. Ugur, "Learning social navigation from demonstrations with deep neural networks,"

[80] O. A. I. Ramírez, H. Khambhaita, R. Chatila, M. Chetouani, and R. Alami, "Robots learning how and where to approach people," in *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)*, pp. 347–353, IEEE, 2016.

[81] S. Sun, X. Zhao, Q. Li, and M. Tan, "Inverse reinforcement learning-based time-dependent a* planner for human-aware robot navigation with local vision," *Advanced Robotics*, vol. 34, no. 13, pp. 888–901, 2020.

[82] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "Socnavbench: A grounded simulation testing framework for evaluating social navigation," *ACM Transactions on Human-Robot Interaction*, vol. 11, no. 3, pp. 1–24, 2022.

[83] M. M. Neggers, R. H. Cuijpers, P. A. Ruijten, and W. A. IJsselsteijn, "The effect of robot speed on comfortable passing distances," *Frontiers in Robotics and AI*, vol. 9, p. 915972, 2022.

[84] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *arXiv preprint arXiv:2103.05668*, 2021.

[85] D. Xie, T. Shu, S. Todorovic, and S.-C. Zhu, "Learning and inferring "dark matter" and predicting human intents and trajectories in videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 7, pp. 1639–1652, 2017.

[86] H. C. Manual, "Special report 209," *Transportation Research Board, Washington, DC*, vol. 1, p. 985, 1985.

[87] I. Ulrich and J. Borenstein, "Vfh+: Reliable obstacle avoidance for fast mobile robots," in *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*, vol. 2, pp. 1572–1577, IEEE, 1998.

[88] S. Bitgood and S. Dukes, "Not another step! economy of movement and pedestrian choice point behavior in shopping malls," *Environment and behavior*, vol. 38, no. 3, pp. 394–405, 2006.

[89] R. McNeill Alexander, "Energetics and optimization of human walking and running: the 2000 raymond pearl memorial lecture," *American journal of human biology*, vol. 14, no. 5, pp. 641–648, 2002.

[90] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT press, 1994.

[91] S. Sagratella, "Algorithms for generalized potential games with mixed-integer variables," *Computational Optimization and Applications*, vol. 68, no. 3, pp. 689–717, 2017.

[92] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26(3), pp. 655–664, Wiley Online Library, 2007.

[93] A. P. Saygin, I. Cicekli, and V. Akman, "Turing test: 50 years later," *Minds and machines*, vol. 10, no. 4, pp. 463–518, 2000.

[94] R. Likert, "A technique for the measurement of attitudes.," *Archives of psychology*, 1932.

[95] B. Prajapati, M. Dunne, and R. Armstrong, "Sample size estimation and statistical power analyses," *Optometry today*, vol. 16, no. 7, pp. 10–18, 2010.

[96] E. Erdfelder, F. Faul, and A. Buchner, "Gpower: A general power analysis program," *Behavior research methods, instruments, & computers*, vol. 28, no. 1, pp. 1–11, 1996.

[97] G. W. Corder and D. I. Foreman, *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons, 2014.

[98] M. Roberts and R. Russo, *A student's guide to analysis of variance*. Routledge, 2014.

[99] M. Kraska-Miller, *Nonparametric statistics for social and behavioral sciences*. CRC Press, 2013.

[100] J. L. Gastwirth, Y. R. Gel, and W. Miao, "The impact of levene's test of equality of variances on statistical theory and practice," *Statistical Science*, vol. 24, no. 3, pp. 343–360, 2009.

[101] G. C. Foster, D. Lane, D. Scott, M. Hebl, R. Guerra, D. Osherson, and H. Zimmer, *An introduction to psychological statistics*. Rice University, 2018.

[102] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[103] M. P. Fay and M. A. Proschan, "Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules," *Statistics surveys*, vol. 4, p. 1, 2010.

[104] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "Socnavbench: A grounded simulation testing framework for evaluating social navigation," *arXiv preprint arXiv:2103.00047*, 2021.

[105] E. Ostertagova, O. Ostertag, and J. Kováč, "Methodology and application of the kruskal-wallis test," in *Applied Mechanics and Materials*, vol. 611, pp. 115–120, Trans Tech Publ, 2014.

[106] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.

[107] D. H. Johnson, "The insignificance of statistical significance testing," *The journal of wildlife management*, pp. 763–772, 1999.

[108] R. Kirby, R. Simmons, and J. Forlizzi, "Companion: A constraint-optimizing method for person-acceptable navigation," in *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 607–612, IEEE, 2009.

[109] J. Müller, C. Stachniss, K. O. Arras, and W. Burgard, "Socially inspired motion planning for mobile robots in populated environments," in *Proc. of International Conference on Cognitive Systems*, 2008.

[110] Y. C. Pradeep, Z. Ming, M. Del Rosario, and P. C. Chen, "Human-inspired robot navigation in unknown dynamic environments," in *2016 IEEE International Conference on Mechatronics and Automation*, pp. 971–976, IEEE, 2016.

[111] Y. Tamura, P. Dai Le, K. Hitomi, N. P. Chandrasiri, T. Bando, A. Yamashita, and H. Asama, "Development of pedestrian behavior model taking account of intention," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 382–387, IEEE, 2012.

[112] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.

[113] C. Dogbé, "Modeling crowd dynamics by the mean-field limit approach," *Mathematical and Computer Modelling*, vol. 52, no. 9-10, pp. 1506–1520, 2010.

[114] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani, "Forecasting interactive dynamics of pedestrians with fictitious play," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 774–782, 2017.

[115] S. Hoogendoorn and P. HL Bovy, "Simulation of pedestrian flows by optimal control and differential games," *Optimal control applications and methods*, vol. 24, no. 3, pp. 153–172, 2003.

[116] Y. Rahmati and A. Talebpour, "Learning-based game theoretical framework for modeling pedestrian motion," *Physical Review E*, vol. 98, no. 3, p. 032312, 2018.

[117] X. Chen, M. Treiber, V. Kanagaraj, and H. Li, "Social force models for pedestrian traffic–state of the art," *Transport Reviews*, vol. 38, no. 5, pp. 625–653, 2018.

[118] A.-H. Olivier, A. Marin, A. Crétual, A. Berthoz, and J. Pettré, "Collision avoidance between two walkers: Role-dependent strategies," *Gait & posture*, vol. 38, no. 4, pp. 751–756, 2013.

[119] F. Zanlungo, T. Ikeda, and T. Kanda, "Social force model with explicit collision prediction," *Europhysics Letters*, vol. 93, no. 6, p. 68005, 2011.

[120] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1688–1694, IEEE, 2013.

[121] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo, "Walking ahead: The headed social force model," *PloS one*, vol. 12, no. 1, p. e0169734, 2017.

[122] E. Cristiani, B. Piccoli, and A. Tosin, "Multiscale modeling of granular flows with application to crowd dynamics," *Multiscale Modeling & Simulation*, vol. 9, pp. 155–182, Jan. 2011.

[123] F. T. Johora and J. P. Müller, "Modeling interactions of multimodal road users in shared spaces," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3568–3574, IEEE, 2018.

[124] P. P. Menon, J. Kim, D. G. Bates, and I. Postlethwaite, "Clearance of nonlinear flight control laws using hybrid evolutionary optimization," *IEEE transactions on evolutionary computation*, vol. 10, no. 6, pp. 689–699, 2006.

[125] L. Moreno, S. Garrido, D. Blanco, and M. L. Munoz, "Differential evolution solution to the slam problem," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 441–450, 2009.

[126] S. Aydin and H. Temeltas, "Fuzzy-differential evolution algorithm for planning time-optimal trajectories of a unicycle mobile robot on a predefined path," *Advanced Robotics*, vol. 18, no. 7, pp. 725–748, 2004.

[127] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *Journal of global optimization*, vol. 27, pp. 105–129, 2003.

[128] R. Storn, "Differrential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces," *Technical report, International Computer Science Institute*, vol. 11, 1995.

[129] A. Johansson, D. Helbing, and P. K. Shukla, "Specification of the social force pedestrian model by evolutionary adjustment to video tracking data," *Advances in complex systems*, vol. 10, no. supp02, pp. 271–288, 2007.

[130] "Tensorflow." https://www.tensorflow.org/overview?hl=it. Accessed: 2023-08-23.

[131] "Ros move_base package." http://wiki.ros.org/move_base. Accessed: 2023.

[132] "Gazebo simulator." https://gazebosim.org/home. Accessed: 2023.

[133] "gazebo_sfm_plugin." https://github.com/robotics-upo/gazebo_sfm_plugin. Accessed: 2023.

[134] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PloS one*, vol. 5, no. 4, p. e10047, 2010.

[135] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds," *Proceedings of the Royal Society B: Biological Sciences*, vol. 276, pp. 2755–2762, 2009.

[136] "Rviz docs." http://wiki.ros.org/rviz. Accessed: 2023.

[137] X.-T. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 4, pp. 1743–1760, 2017.

[138] "Interbotics locobot overview." https://www.trossenrobotics.com/locobot-base.aspx. Accessed: 2023.

[139] "Create3 mobile base overview." https://iroboteducation.github.io/create3_docs/hw/overview/. Accessed: 2023.

[140] "Interbotics locobot specification." https://docs.trossenrobotics.com/interbotix_xslocobots_docs/specifications.html. Accessed: 2023.

[141] "Widowx-250 6dof." https://www.trossenrobotics.com/docs/interbotix_xsarms/specifications/wx250s.html. Accessed: 2023.

[142] "Intel® realsense™ depth camera d435." https://www.intelrealsense.com/depth-camera-d435/. Accessed: 2023.

[143] "Zed2 stereolabs docs: Depth sensing overview." https://www.stereolabs.com/docs/depth-sensing/. Accessed: 2023.

[144] "Zed2 stereolabs docs: Camera controls." https://www.stereolabs.com/docs/video/camera-controls/. Accessed: 2023.

[145] "Zed2 positional tracking overview | stereolabs." https://www.stereolabs.com/docs/positional-tracking/. Accessed: 2023.

[146] "Spatial mapping overview." https://www.stereolabs.com/docs/spatial-mapping/. Accessed: 2023.

[147] "Zed2 3d object detection overview | stereolabs." https://www.stereolabs.com/docs/object-detection/. Accessed: 2023.

[148] "Zed2 stereolabs docs: Api reference, tutorials, and integration." https://www.stereolabs.com/docs/. Accessed: 2023.

[149] C. M. Carpinella, A. B. Wyman, M. A. Perez, and S. J. Stroessner, "The robotic social attributes scale (rosas) development and validation," in *Proceedings of the 2017 ACM/IEEE International Conference on human-robot interaction*, pp. 254–262, 2017.

[150] R. Chin and B. Y. Lee, *Principles and practice of clinical trial medicine*. Elsevier, 2008.

[151] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[152] "Preferred walking speed." https://en.wikipedia.org/wiki/Preferred_walking_speed. Accessed: 2023.

[153] M. M. De Graaf and S. B. Allouch, "Exploring influencing variables for the acceptance of social robots," *Robotics and autonomous systems*, vol. 61, no. 12, pp. 1476–1486, 2013.

[154] Á. Castro-González, H. Admoni, and B. Scassellati, "Effects of form and motion on judgments of social robots' animacy, likability, trustworthiness and unpleasantness," *International Journal of Human-Computer Studies*, vol. 90, pp. 27–38, 2016.

[155] "Average human height by country." https://en.wikipedia.org/wiki/Average_human_height_by_country. Accessed: 2023.

[156] C. R. Crowell, J. C. Deska, M. Villano, J. Zenk, and J. T. Roddy Jr, "Anthropomorphism of robots: Study of appearance and agency," *JMIR human factors*, vol. 6, no. 2, p. e12629, 2019.

[157] J. K. Verma and V. Ranga, "Multi-robot coordination analysis, taxonomy, challenges and future scope," *Journal of intelligent & robotic systems*, vol. 102, no. 1, pp. 1–36, 2021.

[158] M. Mansouri, F. Pecora, and P. Schüller, "Combining task and motion planning: Challenges and guidelines," *Frontiers in Robotics and AI*, p. 133, 2021.

[159] Y. Jiang, J. Hu, and D. Lin, "Decision making of networked multiagent systems for interaction structures," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1107–1121, 2011.

[160] D. Ye, M. Zhang, and D. Sutanto, "Self-adaptation-based dynamic coalition formation in a distributed agent network: A mechanism and a brief survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 1042–1051, 2013.

[161] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 138–153, 2004.

[162] S. A. Zanlongo, P. Dirksmeier, P. Long, T. Padir, and L. Bobadilla, "Scheduling and path-planning for operator oversight of multiple robots," *Robotics*, vol. 10, no. 2, p. 57, 2021.

[163] M. E. Gustavsson, F. K. Arnberg, N. Juth, and J. von Schreeb, "Moral distress among disaster responders: what is it?," *Prehospital and disaster medicine*, vol. 35, no. 2, pp. 212–219, 2020.

[164] M. Harbers, J. de Greeff, I. Kruijff-Korbayová, M. A. Neerincx, and K. V. Hindriks, "Exploring the ethical landscape of robot-assisted search and rescue," in *A World with Robots: International Conference on Robot Ethics: ICRE 2015*, pp. 93–107, Springer, 2017.

[165] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[166] O. Arslan and P. Tsiotras, "Use of relaxation methods in sampling-based algorithms for optimal motion planning," in *2013 IEEE International Conference on Robotics and Automation*, pp. 2421–2428, IEEE, 2013.

[167] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International journal of robotics research*, vol. 23, no. 9, pp. 939–954, 2004.

[168] J. Bellingham, M. Tillerson, A. Richards, and J. P. How, "Multi-task allocation and path planning for cooperating uavs," in *Cooperative control: models, applications and algorithms*, pp. 23–41, Springer, 2003.

[169] M. Otte, M. J. Kuhlman, and D. Sofge, "Auctions for multi-robot task allocation in communication limited environments," *Autonomous Robots*, vol. 44, no. 3, pp. 547–584, 2020.

[170] L. Liu and D. A. Shell, "Optimal market-based multi-robot task allocation via strategic pricing.," in *Robotics: Science and Systems*, vol. 9, pp. 33–40, 2013.

[171] G. P. Das, T. M. McGinnity, S. A. Coleman, and L. Behera, "A distributed task allocation algorithm for a multi-robot system in healthcare facilities," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 1, pp. 33–58, 2015.

[172] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–31, 2019.

[173] K. C. Tan, M. Jung, I. Shyu, C. Wan, and R. Dai, "Motion planning and task allocation for a jumping rover team," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5278–5283, IEEE, 2020.

[174] Z. Li, X. Li, *et al.*, "Research on model and algorithm of task allocation and path planning for multi-robot," *Open Journal of Applied Sciences*, vol. 7, no. 10, p. 511, 2017.

[175] M. Hussain, B. Kimiaghalam, A. Ahmedzadeh, A. Homaifar, and B. Sayyarodsari, "Multi-robot scheduling using evolutionary algorithms," in *Proceedings of the 5th Biannual World Automation Congress*, vol. 13, pp. 233–238, IEEE, 2002.

[176] M. Nanjanath and M. Gini, "Repeated auctions for robust task execution by a robot team," *Robotics and Autonomous Systems*, vol. 58, no. 7, pp. 900–909, 2010.

[177] M. Nanjanath and M. Gini, "Dynamic task allocation for robots via auctions," in *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pp. 2781–2786, IEEE, 2006.

[178] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[179] S. Al-Hussaini, J. M. Gregory, and S. K. Gupta, "Generating task reallocation suggestions to handle contingencies in human-supervised multi-robot missions," *IEEE Transactions on Automation Science and Engineering*, 2023.

[180] S. Öztürk and A. E. Kuzucuoğlu, "Optimal bid valuation using path finding for multi-robot task allocation," *Journal of intelligent manufacturing*, vol. 26, pp. 1049–1062, 2015.

[181] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[182] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics and Automation Magazine*, vol. 19, pp. 72–82, December 2012.

[183] S. Shriyam and S. K. Gupta, "Task assignment and scheduling for mobile robot teams," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 51807, p. V05AT07A075, American Society of Mechanical Engineers, 2018.

[184] L. Cao, H. shun Tan, H. Peng, and M. cong Pan, "Multiple uavs hierarchical dynamic task allocation based on pso-fsa and decentralized auction," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 2368–2373, 2014.

[185] S. Tanha, S. Dehkordi, and A. Korayem, "Control a mobile robot in social environments by considering human as a moving obstacle," in *2018 6th RSI International Conference on Robotics and Mechatronics (IcRoM)*, pp. 256–260, IEEE, 2018.

[186] S. Primatesta, G. Guglieri, and A. Rizzo, "A risk-aware path planning strategy for uavs in urban environments," *Journal of Intelligent & Robotic Systems*, vol. 95, pp. 629–643, 2019.

[187] S. Primatesta, L. S. Cuomo, G. Guglieri, and A. Rizzo, "An innovative algorithm to estimate risk optimum path for unmanned aerial vehicles in urban environments," *Transportation research procedia*, vol. 35, pp. 44–53, 2018.