Modelling of Machine-Aided Human Group Motion

Thomas Given-Wilson, Axel Legay, Sean Sedwards, Olivier Zendra

Modelling of Machine-Aided Human Group Motion

Thomas Given-Wilson
Inria
Rennes, France
thomas.given-wilson@inria.fr

Axel Legay
Inria
Rennes, France
axel.legay@inria.fr

Sean Sedwards
Inria
Rennes, France
sean.sedwards@uwaterloo.ca

Olivier Zendra
Inria
Rennes, France
olivier.zendra@inria.fr

ABSTRACT

The ACANTO project is developing robotic assistants to aid the mobility and recovery of mobility-impaired and older adults. One key feature of the project's robotic assistants is aiding with navigation in chaotic environments. Prior work has solved this for a single user with a single robot, however for therapeutic outcomes ACANTO supports social groups and group activities. Thus these robotic assistants must be able to efficiently support groups of users walking together. This requires an efficient navigation solution that can handle large numbers of users, maintain (de-facto) group cohesion despite unpredictable behaviours, and operate rapidly on embedded devices. We address these challenges by: using sensor information to develop behavioural traces, clustering traces to determine groups, modeling the groups using the social force model, and finding an optimal navigation solution using statistical model checking. The new components of this solution are validated on the ETH Zürich dataset of pedestrians in an open environment.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

human robot interaction, group planning, pedestrian behaviour, social force model, collaborative planning

ACM Reference Format:
Thomas Given-Wilson, Axel Legay, Sean Sedwards, and Olivier Zendra. 2018. Modelling of Machine-Aided Human Group Motion. In *Proceedings of Human Robot Interaction.* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

The ACANTO EU project [1] is developing robotic assistants to aid mobility of mobility-impaired and elderly adults. These robotic assistants provide a variety of support to their users, including: navigational assistance, social networking, social activity planning, therapeutic regime support, and diagnostic support.

The navigational assistance and social activities are the focus in this paper, as together they yield an interesting challenge in human robot interaction. The goal is to help groups of users navigate in a potentially busy dynamic environment, while also maintaining social group cohesion.

A robotic assistant has been developed before in the DALi project [9], acting selfishly to ensure the safe navigation of a single user. This was achieved by using the social force model and statistical model checking in a reactive planner that frequently replanned and made immediate navigational suggestions to the user. The key operational loop of this solution was to: observe the environment, model the agents in the environment in the social force model, give safety constraints for the user, and then use statistical model checking to find the optimal next move for the user.

Generalising to groups of users poses several significant difficulties. *Computationally*, the challenge is exponential in the number of users, considering all their possible navigational choices. *Incomplete information* is normal, since sensors are distributed between robotic assistants and the environment, and communication may fail, leading to different robots having different knowledge of the environment. *Maintaining group cohesion* is non-trivial, since group composition and position are dynamic and, unlike swarm robotics, no group member can be abandoned. *Frequent replanning* is necessary since there is minimal control over the users' actions, which may include ignoring the advise of the robotic assistant.

The solution proposed here is to *abstract away from individual users in favour of groups*. This refines the prior solution for a single user. Sensor information is used to obtain *traces* that provide behavioural information about users and pedestrians in the environment. These traces are clustered into groups that capture both location and motion behaviour. The groups are used as the social particles in the social force model, with parameters adjusted to account for group dynamics. Statistical model checking is used to find the optimal next move for the group containing the user, and the navigation for the optimal next move is displayed to the user.

The effectiveness of the group abstraction mechanisms use in this refined algorithm are validated on the ETH Zürich BIWI walking pedestrians dataset [12]. This shows they operate correctly and effectively, even improving over human annotations, on real world data of pedestrians in a chaotic environment.

The rest of the paper is structured as follows. Section 2 recalls background tools. Section 3 presents the context of the ACANTO

Human Robot Interaction, March 2018, Chicago, Illinois USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

project and explains in more details the challenge faced. Section 4 presents our solution for group motion planning. Section 5 validates the algorithms using the ETH Zürich pedestrian data set. Section 6 concludes and considers future work.

2 BACKGROUND

This section briefly describes the social force model (SFM) and statistical model checking (SMC) for later use.

2.1 The Social Force Model

The SFM [16–19] models the social and physical interactions of pedestrian agents as explicit forces. Denoting vectors in bold type, agent i has mass m_i centered at position $\mathbf{x}_i \in \mathbb{R}^2$ in the environment, radius r_i and velocity $\mathbf{v}_i \in \mathbb{R}^2$. The SFM is described by a system of linear differential equations

$$\begin{cases} \dot{\mathbf{x}}_i = \mathbf{v}_i \\ \dot{\mathbf{v}}_i = \frac{\mathbf{v}_i^0 - \mathbf{v}_i}{\tau_i} + \frac{\mathbf{f}_i + \xi_i}{m_i} \end{cases} \tag{1}$$

$\mathbf{v}_i^0$ is the *driving (desired) velocity* of agent i, represented by a product of speed v_i^0 and normalised direction $\mathbf{e}_i^0$. Usually, $\mathbf{e}_i^0$ is given by the line joining the current position and the next via point. Importantly, since v_i^0 is by default set to the user's preferred walking speed, $\mathbf{v}_i^0$ is time invariant between via points. τ_i is the time taken to react to the difference between desired and actual velocity, while ξ_i is a noise term modelling fluctuations not accounted for by the deterministic part of the model. The noise term can also serve to avoid deadlocks and hypothesise alternative trajectories. Usually, the ξ_i is assumed normally distributed. In the absence of the exogenous inputs $\mathbf{f}_i$ and ξ_i, the agent's trajectory simply converges to the driving velocity with time constant τ_i. $\mathbf{f}_i$ is the overall force acting on agent i resulting from other objects in the environment and is given by

$$\mathbf{f}_i = \sum_{j \neq i} [\mathbf{f}_{ij}^{\text{soc}} + \mathbf{f}_{ij}^{\text{att}} + \mathbf{f}_{ij}^{\text{ph}}] + \sum_b [\mathbf{f}_{ib}^{\text{soc}} + \mathbf{f}_{ib}^{\text{ph}}] \tag{2}$$

The first term on the right-hand side of (2) includes all the forces on agent i resulting from interactions with other agents: $\mathbf{f}_{ij}^{\text{soc}}$ is the repulsive social force that inhibits strangers from getting too close, $\mathbf{f}_{ij}^{\text{att}}$ is the attractive social force that brings agents together, $\mathbf{f}_{ij}^{\text{ph}}$ is the physical force that exists when two agents come into contact. The second term includes the forces acting on agent i as a result of fixed environmental obstacles: $\mathbf{f}_{ib}^{\text{soc}}$ is the social force that inhibits agent i from getting too close to the boundaries, $\mathbf{f}_{ib}^{\text{ph}}$ is the physical force that exists when agent i touches the boundary b.

$\mathbf{f}$ is principally a function of the distance between an agent and the other objects in the model. d_{ib} is the minimum distance between the circumference of agent i and fixed object b. d_{ij} is the distance between the centres of mass of agents i and j, i.e., the centres of the discs, while $r_{ij} = r_i + r_j$ is the "touching distance". To aid modelling the different force regimes that exist when agents are not in contact and when they touch (i.e. agents i and j touch if $r_{ij} - d_{ij} \leq 0$) it is customary to choose the function $\Theta(r_{ij}, d_{ij}) = \max(0, r_{ij} - d_{ij})$.

Using these notions, the various repulsive social and physical forces of (2) are defined as follows:

$$\mathbf{f}_{ij}^{\text{soc}} = \left\{A_i \exp[(r_{ij} - d_{ij})/B_i]\right\} \mathbf{n}_{ij} \Lambda(\lambda_i, \varphi_{ij}) \tag{3}$$

$$\mathbf{f}_{ij}^{\text{ph}} = k_1 \Theta(r_{ij} - d_{ij}) \mathbf{n}_{ij} + k_2 \Theta(r_{ij} - d_{ij}) \Delta v_{ji}^t \mathbf{t}_{ij} \tag{4}$$

$$\mathbf{f}_{ib}^{\text{soc}} = \{A_i \exp[(r_i - d_{ib})/B_i] + k_1 \Theta(r_i - d_{ib})\} \mathbf{n}_{ib} \tag{5}$$

$$\mathbf{f}_{ib}^{\text{ph}} = -k_2 \Theta(r_i - d_{ib})(\mathbf{v}_i \cdot \mathbf{t}_{ib}) \mathbf{t}_{ib} \tag{6}$$

$\mathbf{n}_{ij}$ ($\mathbf{n}_{ib}$) is a normalised vector pointing from agent j (fixed object b) to agent i, i.e., the direction of the repulsive force. $\mathbf{t}_{ij}$ ($\mathbf{t}_{ib}$) is a normalised vector tangential to the relative movement of agent i and agent j (fixed obstacle b), i.e., the motion tangential direction. $\Delta v_{ji}^t = (\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{t}_{ij}$ is the tangential velocity difference. The social forces (3) and (5) increase exponentially with reducing distance between objects, with a scale defined by constants A_i and B_i. In particular, A_i is the force acting on agent i at the touching distance; B_i is loosely the distance at which the force takes effect.

$\Lambda : \mathbb{R}^2 \mapsto [0, 1]$ is a function that gives greater weight to the social force (3) arising from the agents in front of (notionally, *seen* by) an agent. λ_i is a parameter that regulates the effect of Λ on agent i, while φ_{ij} is the angle between the directions $\mathbf{e}_i^0$ and $-\mathbf{n}_{ij}$, i.e., the field of view of the agent. The physical force (4) between agents comprises a repulsive body compression force (first term) that acts in direction $\mathbf{n}_{ij}$, plus a frictional force (second term) that acts in direction $\mathbf{t}_{ij}$ to impede the relative tangential movement of two agents in contact. k_1 and k_2 are constants that define the scale of the physical forces. The physical force (6) between an agent and a fixed object is solely described by a frictional term.

For further discussion and applications of SFM, including variations that account for more human like motion, the reader is referred to [13, 14].

2.2 Statistical Model Checking

Statistical model checking (SMC [20, 31]) is a variety of probabilistic model checking that avoids an explicit representation or traversal of the state space and estimates the probability of a property from an empirical distribution of executions (simulations) of the system. Given a number of statistically independent simulation traces of a stochastic model and an automaton to decide whether a trace satisfies a property, it is possible to estimate (with known confidence) the probability that the model will satisfy the property.

For example, the sequential probability ratio test [30, 31] efficiently evaluates the truth of an hypothesis without needing to calculate the actual probability, while the Okamoto bound [20, 27] assigns a level of confidence to a probability estimate for a given number of simulations.

An efficient implementation of SMC is the PLASMA-lab SMC library [4, 28], which checks properties of traces expressed in bounded linear temporal logic (BLTL). This logic can be used to construct complex behavioural properties that express nested temporal causality.

For reference, the typical abstract syntax of a BLTL property is as follows:

$$\phi = \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \mathrm{F}_{\leq t}\phi \mid \mathrm{G}_{\leq t}\phi \mid \phi \,\mathrm{U}_{\leq t}\phi \mid \mathrm{X}_{\leq t}\phi \mid \alpha \tag{7}$$

$\vee$, $\wedge$ and $\neg$ are the standard logical connectives *or*, *and* and *not*, while α is an atomic property that is *true* or *false* in a given state. In practice, α may be an expression over state variables or externally derived metrics of the trajectories. X, F, G and U are temporal operators with a parameter of t time units (real time or a number of steps). X is the *next* operator: $X_{\leq t}\phi$ asserts that ϕ will be true at time / step t. F is the *finally* or *eventually* operator: $F_{\leq t}\phi$ asserts that ϕ will be true at some time within t time units. G is the *globally* or *always* operator: $G_{\leq t}\phi$ asserts that ϕ will be true at all times within t time units. U is the *until* operator: $\psi U_{\leq t}\phi$ asserts that ϕ will be true within t time units and ψ will be true until it is.

It is straightforward to express properties important to ACANTO using BLTL. For example, if ϕ represents collision between the user and any other agent, then $G_{\leq\infty}\phi$ represents never allowing a collision to occur. Similarly, by ensuring encoding in ϕ that the user is never isolated from their group could be represented by $G_{\leq\infty}\phi$. Other properties can be encoded using other operators, e.g.: if ϕ represents reaching the destination, then $F_{\leq t}\phi$ ensures the destination is reached by time t, and if ϕ' represents being within close proximity to a bathroom then $\phi' U_{\leq t}\phi$ ensures the user is close to a bathroom until their destination is reached.

BLTL's expressiveness as well as the ability to model continuous time and space, uncertainty about actions and information, all make SMC an excellent choice here. Further, the requirement of a quick approximately best answer, rather than a certain answer eventually (see Section 3), also make SMC an ideal technology to use.

3 CONTEXT AND MOTIVATION

This section overviews key aspects of the ACANTO project and the challenge of collaborative machine-aided human group motion. This includes the ACANTO reactive planner, the challenge constraints, and related works in group robotic navigation.

The ACANTO robotic assistant *FriWalk* is a "walker" device equipped with wheels, sensors, motors, and brakes that helps mobility-impaired humans walk by providing physical support. Further, the FriWalk is equipped with a *FriTab*, an embedded computer with a touch-screen interface for the user. The FriTab integrates with the FriWalk sensors, as well as the larger ACANTO infrastructure, to provide navigational assistance for the user. This assistance is a combination of both long term navigation and trip planning, and short term collision avoidance and traffic flow. For simplicity, and where no ambiguity may occur, the term *walker* shall be used in this paper to refer to a FriWalk with integrated FriTab. The term *user* shall refer to a human using a walker, while *pedestrian* shall be used for all pedestrians (including users).

3.1 Reactive Planner

The navigational challenge of the walker is to gather local environment information and provide an immediate and safe navigational suggestion to the user. This is handled by the *reactive planner*, which operates on a fast local loop and can react to the potentially chaotic environment rapidly and update navigation assistance advice. The reactive planner component was effective in the DALi project, so the same architecture was also chosen for ACANTO.

The reactive planner architecture for DALi and ACANTO is shown in Figure 1. Sensors provide information that is processed and passed to locate the user, fixed objects and pedestrians in the local environment. Then the reactive planner suggests an instantaneous direction that approximately maximises the probability of satisfying the user's objectives.

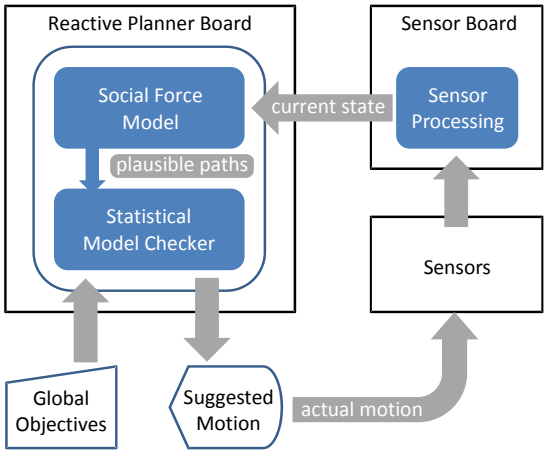

Figure 1: Reactive Planner architecture.

Algorithm 1: DALi Reactive Planner

```
Let Move be the set of possible navigation suggestions
Let Cstr be the set of constraints in BLTL
Let Self be the current position and velocity
while active do
    Let Obs be the set of current observations
    Set action ← stop
    Set prob_success ← 0
    for m ∈ Move do
        Set model ← SFM(Obs∪(Self, m))
        Set prob_m ← SMC(Cstr, model)
        if prob_m > prob_success then
            action ← m
            prob_success ← prob_m
    Display action to user.
```

Algorithm 1 describes the function of the DALi reactive planner. The position and (possibly zero) velocity of objects in the environment *Obs* are identified by the sensors and used to parametrise the human motion model, here the SFM represented by *model* set to $SFM(Obs \cup (Self, m))$. The SMC simulates multiple future trajectories of the model up to some time horizon, given different hypothesised initial directions m of the user. Each trajectory is validated against the user's constraints *Cstr* and thus the planner is able to estimate the probability of success $prob_m$ for each hypothesised direction m. The direction suggested to the user *action* is that which maximises the probability of success ($prob_{success}$) and minimises deviation from a direct path to the next waypoint.

This algorithm forms the basis for the solution here, but is enhanced to account for the challenges of ACANTO (see Section 4.3).

3.2 Challenges

For a single user with a single walker the DALi solution is proven successful [8, 9]. This holds even though the user is free to ignore or override the navigational advice at any time. Thus, the rôle of the walker is to advise the user. Distinct from most robotic systems, the walker cannot freely choose its actions and instead has its movements chosen by the user.

In ACANTO, the navigational challenges are generalised to social groups that may have many pedestrians and users participating in an activity together. This implies various challenges since the walker must now act cohesively and cooperatively in a group, including with other walkers and group members who do not have walkers. This raises several difficulties, for which the single-user approach of the DALi project is no longer immediately feasible.

Computationally, the problem becomes exponential in the number of users. This is due to modelling all possible directional choices for all users (i.e. Algorithm 1 becomes $|Move|^n$ where n is the number of users).

Incomplete information is inherent since sensors are distributed and communication cannot be guaranteed. Further, a walker cannot (computationally) model the entire environment, and so must limit the range of sensor information considered. Thus, different walkers will have different knowledge and incomplete knowledge. In practice *Obs* in Algorithm 1 will differ between walkers.

Maintaining group cohesion is non-trivial, since groups may be ad-hoc rather than fixed, and identification of members of a group must be dynamically computed. Even given the same information, Algorithm 1 only optimises for *Self*. This is of particular importance to social groups, where the social goals of global activity plans will fail if group cohesion is lost.

Frequent replanning becomes necessary from the above constraints and the chaotic environment, since otherwise navigation will rapidly be out of date.

These difficulties specific to the ACANTO context make using a classical (DALi) solution infeasible.

3.3 Related Works

This section contrasts the challenge with related works on group robotic motion planning, recalling how these are infeasible for ACANTO.

Classical robotics solutions fail to account for some of the above difficulties and the advisory rôle of the walker, and so cannot be easily applied. In classical robotics, the robots are typically under full control of the navigation choices so they can be forced to follow the plan or actions deemed most likely to achieve the goal [3, 6, 7, 11, 29]. Even in the case of disruptions or problems, robots can be forced to take a particular path, or wait indefinitely until it is "safe" to proceed [6, 7, 29]. By contrast, human users may override navigational advise, make mistakes, or pursue their own goals, unknown to the navigation engine.

Even compared to robotic systems that must consider disruptions [11, 29], the variability of the environment and disruptions in the ACANTO context may be significantly higher, since pedestrians are humans in an open pedestrian environment, and the users are free to act as they please, including ignoring the walker's advice.

Although this scenario is superficially similar to swarm robotics, in the sense of having multiple agents with a common goal, the success criteria are very distinct. Typically in swarm robotics success is achieved when some number of robots reach the goal or find a path [10, 23–26, 32], however for ACANTO an activity is only considered completely successful if all the social group members complete the activity, and do so coherently.

4 GROUP MOTION PLANNING

This section presents the technology and algorithms used to perform navigational support and advise in ACANTO.

The solution here improves on the DALi reactive planner (see Section 3.1) to address the unique challenges of ACANTO (see Section 3.2). The key concept of the solution is to abstract away form individuals in favour of groups in the reactive planning algorithm (i.e. replace Algorithm 1), and to exploit improved information from greater number of users and sensors.

Abstracting to groups naturally captures group behaviours and dynamics, and navigation at the level of groups solves the difficulties of attempting to do individual navigation and have all the individuals in a group maintain group coherency.

For improved behaviour prediction it is necessary to know the intentions of pedestrians, i.e. where they're trying to go and with whom they're travelling. This information is well defined for participants of an ACANTO activity, but not for unknown pedestrians. Therefore past behaviour is observed to infer future behaviour, represented by a pedestrian's *trace*.

Combining the output of sensors on multiple walkers incurs the additional challenge of identifying pedestrians who leave the view of one sensor and appear in another. Pedestrians may also appear and disappear as a result of sensors being obscured, because of communication unreliability or because users just leave. Thus the trace inference algorithm, described in Section 4.1, makes use of a human motion model to reliably link the *behaviour* of pedestrians over much longer times than that between successive (sensor) video frames.

Having inferred a set of traces from multiple observations of users in the environment, it is necessary to infer their de-facto groups. Instantaneous physical proximity is not a sufficient indicator, since two close pedestrians could actually be trying to get away from each other. The group inference algorithm (Algorithm 3) thus uses a notion of proximity that includes both position and velocity: if pedestrians are physically close, walking at the same speed in the same direction, it is reasonable to assume (by definition) that they are walking together. Finding an optimal partition of traces into groups here uses k-means clustering [22]. The full technical aspects of this are outlined in Section 4.2.

Identifying de-facto groups allows the walker to plan motion at a more efficient level of abstraction. When hypothesising the alternative directions for a number of users of the platform, it is a reasonable compromise to only hypothesise the overall motion of the groups to which users belong. It is not necessary to consider all the possible combinations of suggestions to those within the same group, given that, by virtue of how a group is defined, their motion is strongly correlated. Note that suggestions are nevertheless tailored

to the actual position of a user within the group, in order to maintain its "social" structure.

The remainder of this section details specific algorithms used to achieve these recommendations for users to maintain safety and social group goals.

4.1 Trace Inference

The trace inference algorithm (Algorithm 2) constructs sets of active and inactive traces, where a trace is a sequence of time-stamped observations of the position and velocity of pedestrians detected by the sensors. Active traces are those for which the algorithm has reliably inferred continuity and/or there is currently a pedestrian in the field of view of the sensors (a trace may consist of a single observation). Inactive traces are those for which the algorithm could find no valid continuation, so there is no current view of the corresponding pedestrian. Since inactive traces do not contain a current point, further trace inference applies only to active traces. In practice a trace may become inactive due to the obfuscation of a sensor or failure of communications. The algorithm assumes that the human motion model makes valid predictions up to a maximum time interval of Δt_{max}, hence traces whose endpoints are older than this become inactive.

The initial set of active traces comprise the initial set of observations, while the initial set of inactive traces is empty. Each step of the algorithm appends new observations to active traces or starts new active traces with single observations that cannot be assigned to existing active traces. Active traces to which no new point can be assigned become inactive once their endpoint is older than Δt_{max} with respect to the current time. Using the SFM, each iterative step of the algorithm generates a set of projections to the current time from the the endpoints of the active traces, then tries to match the projected points to the new observations by finding a minimum distance assignment according to distance metric D_{trace} (see Section 5.2).

To solve the *assignment problem* of optimally matching new observations to projected points, the implementation makes use of an $O(n^3)$ implementation of the Hungarian method [5, 21]. First a square matrix is constructed of distances between projected points and the new observations, according to distance metric D_{trace}. The numbers of projected points and active traces may be different, so the smaller of the two sets is padded with dummy entries whose distance to all members of the other set is, by convention, made equal to the maximum observed distance between non-dummy entries. The Hungarian method is guaranteed to find an assignment that minimises the overall distance between the projected points and new observations, but not all individual assignments are close enough to be accepted and some assignments include non-existent (dummy) entries. Observations whose assignment has a distance up to threshold θ_{trace} are appended to the ends of the corresponding active traces. Observations whose assignment has a distance greater than θ_{trace} become the initial points of new active traces. Assignments involving dummy entries are discarded.

Algorithm 2: Trace inference

```
Let H be the human motion model
Let Act be the current set of active traces (initialised with
  the first set of observations)
Let Inact be the current set of inactive traces (initially
  empty)
Let Δt_max the maximum trace projection time
Let D_trace be the trace matching distance metric
Let θ_trace be the trace matching threshold distance
while there are new observations do
    Let Obs be the set of new observations at current time t
    Let Old ⊆ Act be the set of traces whose end points are
      older than t − Δt_max
    Act ← Act\Old
    Inact ← Inact ∪ Old
    Let Proj be the set of points generated by projecting the
      ends of all traces in Act to time t using H
    if |Obs| < |Proj| then
        Pad Obs with dummy entries so |Obs| = |Proj|
    else if |Proj| < |Obs| then
        Pad Proj with dummy entries so |Obs| = |Proj|
    Construct a square matrix Dist of distances between Obs
      and Proj using D_trace
    (set the distance to or from any dummy entry to be the
      maximum non-dummy distance)
    Apply the Hungarian Method to Dist to find a minimum
      distance set of assignments Assign
    Remove from Assign all assignments involving dummy
      entries
    for ((trace, projection), observation) ∈ Assign do
        if D_trace(projection, observation) ≤ θ_trace then
            Append observation to trace
        else
            Add observation to Act
```

4.2 Group Abstraction

To infer groups, k-means clustering [22] is used over the set of active traces. The k-means algorithm partitions a set of $n \geq k$ data points into $\leq k$ clusters, according to a specified metric over the points. Although the k-means problem is computationally hard, there are good heuristics that make it expedient for on-the-fly inference application (e.g., the k-means++ algorithm [2]).

Finding an optimal group abstraction is done by first defining a group distance metric, D_{group}, and defining a group cohesion threshold distance, $\theta_{group} > 0$, that specifies the maximum permissible distance from the centroid of a group. Note that the group distance metric is more concerned with similar motion than physical proximity, so being close to the centroid implies primarily that members of the group are moving in the same way. A priori, the optimal number of clusters (i.e., the optimal value of k) is unknown, so the algorithm iterates from $k = 1$ to $k = |Act|$, where Act is the set of traces to cluster, stopping when the set of clusters are sufficiently cohesive. For the purposes of efficient motion planning, minimising k is desirable, so the algorithm aims to find the fewest number of sufficiently cohesive clusters. To judge the cohesiveness of a cluster, the algorithm calculates the distance between each member and the cluster's centroid using D_{group}. If any member

of any cluster is too far from its corresponding centroid, the current set of clusters is abandoned and a new set is generated using $k \leftarrow k + 1$. The algorithm is guaranteed to terminate because when $k = |Act|$, all clusters contain a single element whose distance from their corresponding centroid is guaranteed to be $< \theta_{group}$.

Algorithm 3: Group inference

```
Let Act be the set of active traces
Let D_group be the group distance metric
Let θ_group be the group cohesion threshold distance
Set done ← false
Set k ← 1
while ¬done ∧ k ≤ |Act| do
    Perform k-means clustering on Act
    Let Clust be the resulting set of ≤ k clusters of traces
    done ← true
    for cluster ∈ Clust do
        Let centroid be the centroid of cluster
        for trace ∈ cluster do
            if D_group(trace, centroid) > θ then
                done ← false
return Clust
```

Although the k-means algorithm partitions data points into Voronoi cells that are disjoint in the multi-dimensional space of the distance metric, groups may physically overlap. This arises, for example, when two groups walking in opposite directions pass through each other. This phenomenon does not occur at the level of individuals and is therefore not considered in the original social force model, however it is nevertheless possible to model it with forces in the SFM framework. For example, it is possible to reduce the repulsive social force between the groups ($\mathbf{f}^{\text{soc}}$ in (2)) and use the physical component ($\mathbf{f}^{\text{ph}}$ in (2)) to model the "friction" between them. To accurately model the momentum of different sized groups, the mass term (m in (1)) will be the sum of the masses of the individuals. The latency parameter (τ in (1)) is also likely to be greater for groups, however some of the latency is simply accounted for by the increased mass.

Figure 2 illustrates an hypothetical scenario of groups diverging and coalescing over time, noting that these phenomena are also evident in the automatically-generated visualisations of the output of these on-the-fly algorithms applied to real observations (see Section 5). The groups labelled 1, 2 and 3 are assumed to comprise a single pedestrian. Group 4 contains two pedestrians who initially just happen to be walking close to one another in the same direction. Pedestrians 1 and 3 know each other, so they move closer and eventually form group 5. Pedestrian 2 knows one of the members of group 4, so they also move closer to one another and eventually form group 6. The other member of group 4 is just passing through and eventually leaves the view of the sensors (smaller dashed circle). At some time before then, however, he gets very close to pedestrians 2 and 3 (in the region denoted by the larger dashed circle), but no new group is detected because they are all travelling in different directions. The members of groups 5 and 6 are actually part of the same activity, so the system guides them closer, thus eventually forming group 7.

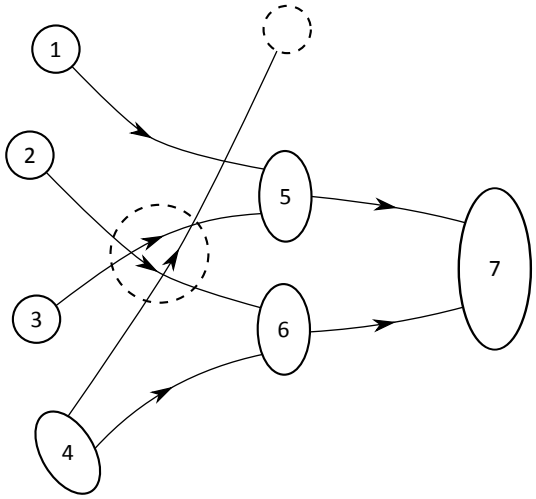

Figure 2: Groups diverging and coalescing.

4.3 Group Motion Planning and Advice

Algorithm 4: ACANTO Reactive Planner

```
Let Move be the set of possible group navigation suggestions
Let Cstr be the set of constraints in BLTL
Let Self be the current user trace
while active do
    Let Obs be the set of current observations
    Let Trace be the set of traces inferred by Algorithm 2
      (using Obs and past state)
    Let Clust be the clusters given by Algorithm 3 (using
      Trace ∪ Self)
    Let Grps be weighted groups derived from Obs and Clust
    Let Sgrp be the group containing Self
    Grps ← Grps \ Sgrp
    Set action ← stop
    Set prob_success ← 0
    for m ∈ Move do
        Set model ← SFM(Grps∪(Sgrp, m))
        Set prob_m ← SMC(Cstr, model)
        if prob_m > prob_success then
            action ← m
            prob_success ← prob_m
    Display action to user.
```

The trace inference and group abstraction can now be added in to the algorithm for the ACANTO reactive planner, as shown in Algorithm 4. Sensor information *Obs* is now converted into traces using Algorithm 2, and this is used to infer groups *Grps* with Agorithm 3. The group containing the user is then separated out and considered as the only group that the possible move suggestions can be given to. The remainder of the algorithm continues as in DALi (Algorithm 1), yielding the best group motion that can match the constraints. Note that here constraints are assumed to also include

group level constraints and so this yields the optimal outcome for the user and their group[1].

The outcome of the SMC engine on the group SFM is then tailored to match the individual user's location within the group. Thus, if the individual is on the fringe of the group they can be guided to rejoin, or similarly avoid trying to merge into a crowded centre of the group. However, such suggestions are minor and the overall advise is the best direction according to the reactive planner/SMC.

5 EXPERIMENTAL VALIDATION

This section presents results of applying the trace inference and group abstraction Algorithms (2 & 3) to the ETH Zürich BIWI walking pedestrians dataset. This demonstrates the effectiveness of the trace inference and group abstraction, thus completing the solution for group motion planning via an improved reactive planner.

5.1 Dataset

The chosen dataset comprises hand-annotated motion-capture observations of pedestrians in two environments: a hotel lobby and a corridor within the ETH premises. The annotations link the observations into traces and groups, which the algorithms presented here do automatically. For the experiments the original annotations are only used to compare with the automated annotations, noting here that this work's algorithms successfully identify all the traces and groups identified by hand. In both environments the observations are made using a fixed camera directly overhead, with an original video frame rate of 25fps. In this paper the focus is on the hotel lobby data, which contains more interesting and varied interactions between pedestrians. In what follows the term 'dataset' refers exclusively to this hotel data.

The dataset contains observations sampled at 2.5fps (0.4s), with observations divided into 27 contiguous intervals separated by more than 0.4s. The excluded times are assumed to be excluded because they contain no *moving* pedestrians, however the results here suggest that some of the omitted frames nevertheless contained *stationary* pedestrians. Moving pedestrians must avoid stationary pedestrians, so for the purpose of motion planning they cannot be ignored. This is illustrated in the results described below. Note that all times are relative to the first frame of the dataset.

5.2 Distance Metrics

The presented results make use of two different distance metrics that are based on both position and velocity. In the case of the trace inference algorithm (Algorithm 2), to decide whether a projected point is "close" to an actual observation, the metric D_{trace} is based on Euclidean distance in the 4-dimensional space of (x, y) position and *discounted* (x, y) velocity. Precisely, the velocity dimension is divided by 2, thus making position more significant than velocity when inferring traces. The value of 2 was chosen empirically and found to work well, but is not critical. Projected points have an implicit velocity that should be similar to that of the observation, but giving too great an emphasis to velocity-matching risks erroneously cross-linking the trajectories of pedestrians walking in formation.

[1]This is not selfish in the sense that the group as a whole has constraints that should all be satisfied. However, a balance must be found between prioritising the individual or the group [15].

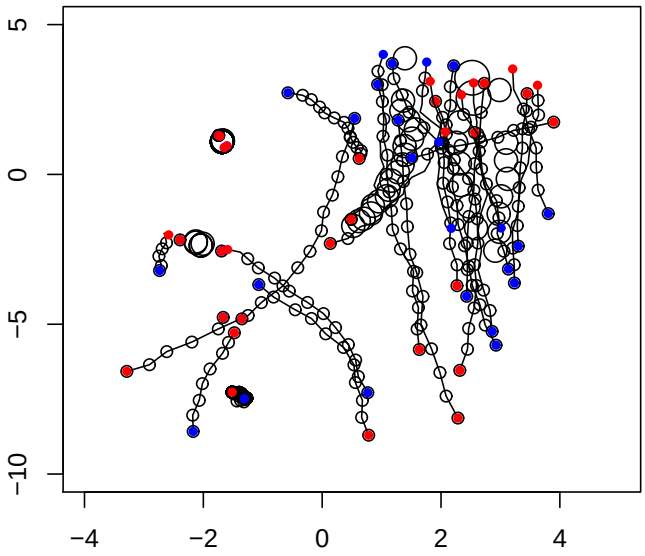

Figure 3: Group abstraction of interval 34.8s to 82.4s.

In the case of the group inference algorithm (Algorithm 3), the metric D_{group} is based on Euclidean distance in the 4-dimensional space of *discounted* (x, y) position and (x, y) velocity. Hence, in contrast to trace inference, velocity is made more significant than position to infer groups using k-means clustering. To justify this group abstraction motion planning, it is required that pedestrians in a group are moving in a similar way, hence velocity is clearly important. It is also required that pedestrians in a group are proximal, but this is less important. A discount factor of 2 was, once again, found empirically to be good and not critical. In fact, very similar results were achieved with no discounting, while usable results were obtained by discounting velocity instead of position.

5.3 Results

Figures 3, 4, 5 & 6 visualise traces and group abstractions for four intervals from the dataset, produced automatically by the algorithms. The x and y axes give the spatial coordinates (in metres) of the groups, with respect to the origin defined in the dataset. The traces inferred by the algorithms are denoted by black lines, marked at their start by blue discs and at their end by red discs. Each black circle denotes an inferred group at a particular time point. A (trivial) group may consist of a single pedestrian (the smallest circles). Most non-trivial groups in these figures consist of pairs of pedestrians (medium-sized circles), with group containing three pedestrians (largest circle) evident in only Figures 3 & 4. Note that the figures abstract away from time, such that the starts and ends of different traces are not necessarily synchronous, hence intersecting traces do not necessarily imply a collision.

Figure 3. The traces and groups identified by automatic algorithms include all those identified in the hand annotation of the original dataset, as well as other groupings that are useful for motion planning. In particular, the automatic algorithms identify stationary groups near coordinates $(-1.7, 1.1)$, $(-2.1, -2.3)$ and $(-1.4, -7.4)$, where the "group" near $(-1.4, -7.4)$ is a trivial group comprising a

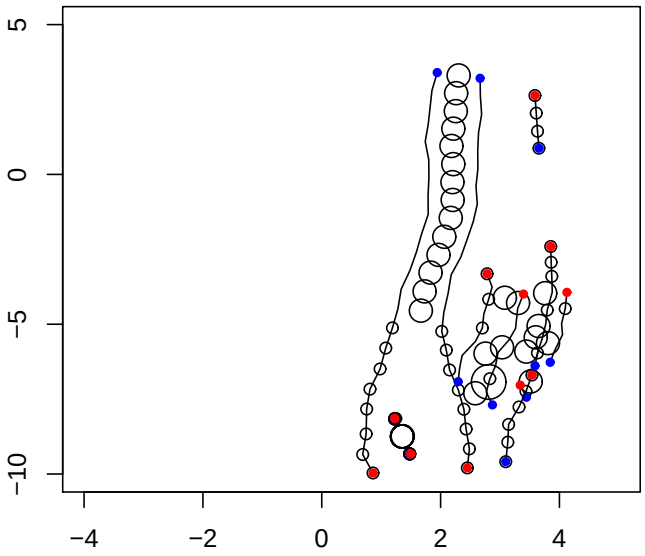

Figure 4: Group abstraction of interval 110.8s to 119.2s.

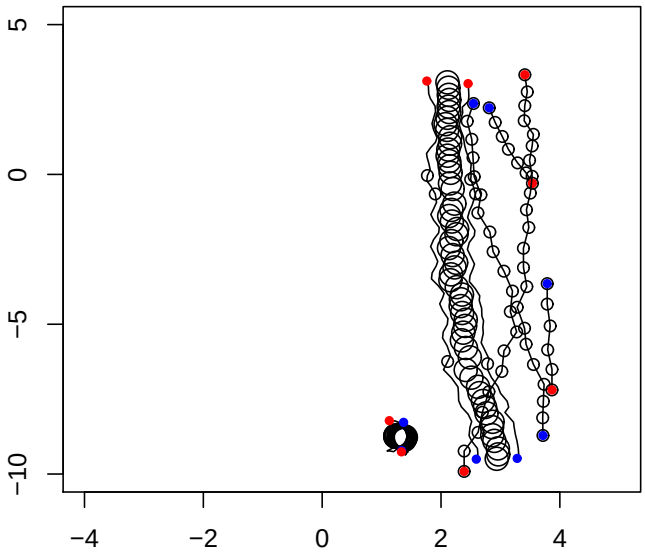

Figure 5: Group abstraction of interval 177.2s to 200.4s.

single pedestrian. This illustrates the effect of stationary groups on moving groups in Figures 4 to 6, which are described below.

Figure 4. First note a stationary group around coordinate $(1.5, -9)$. Not apparent from the figure, they first appear at 113.6s and (approximately) maintain their positions for 5.6s until the end of the interval. Since both their positions and motion are very close, the automatic algorithm correctly identifies these pedestrians as a group. In the original dataset, however, they are not identified as such. Observe that the existence of this stationary group has a significant effect on the motion of the group starting near coordinate $(2.3, 3.3)$. For some reason, the moving group heads directly *towards* the stationary group and then splits near coordinate $(1.7, -4.5)$ to avoid a collision. Importantly, however, up to the point at which the group splits, the group abstraction of the two moving pedestrians provides a good prediction of their behaviour. Following the split, the group abstraction *continues* to provide a good prediction of the moving pedestrians' behaviour because it detects that they are no longer moving together. The other groups in this figure, within the box created by $x \in [2, 4]$ and $y \in [-8, -3]$, are not identified by the hand-annotation of the original dataset, but are nevertheless valid and useful for motion planning.

Figure 5. There is once again a stationary group near coordinate $(2.8, -6.9)$, which is not identified in the hand annotations. Given that this group also exists in the three intervening intervals between those of Figures 4 and 5, it is presumed that this is the same group.

The motion of the group starting near coordinate $(2.9, -9.5)$ is interesting in comparison to the long group trajectories in the other figures. In this figure the pedestrians are walking slower (closer observations made at the same frequency as the other figures), closer together and while not following a smooth path, they appear to very closely maintain their separation distance. It may be possible to infer from this that they have a strong social connection. From the perspective of motion planning they are effectively moving as a single agent, thus fully justifying the group abstraction.

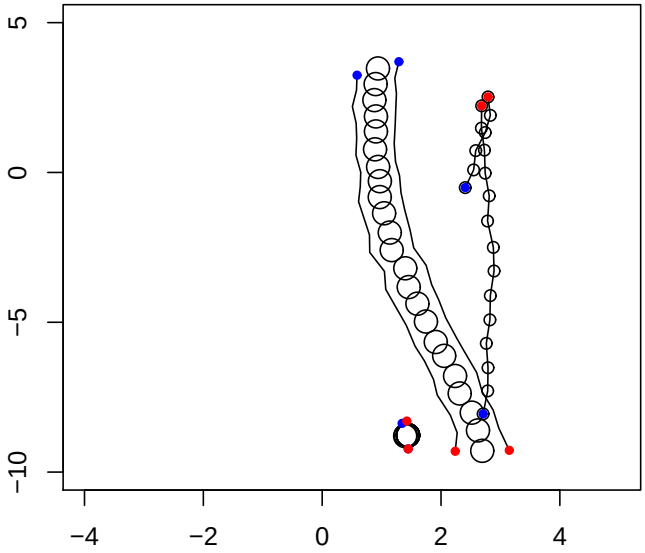

Figure 6: Group abstraction of interval 227.2s to 236s.

Figure 6. The stationary group seen in Figures 4 and 5 seems also to be present in this interval, however the group first appears at time 232.8s, 5.6s after the beginning of the interval, and is not present in the single interval between those illustrated in Figures 5 and 6. From only the recorded observations in the dataset, it is not possible to disambiguate the possibilities that the group moved away and then returned to the same spot, that the camera was temporarily obscured or that there was a data processing error. Temporary occlusions and measurement errors are to be expected, especially when inferring a global view from ground-level cameras mounted on individual walkers, as in ACANTO, and do not invalidate the approach unless the occlusions and errors persist indefinitely.

6 CONCLUSIONS AND FUTURE WORK

The ACANTO project poses a difficult challenge for collaborative robotic navigation assistance and human-robot interaction. That is, to provide navigation advice for mobility-impaired and older adults that must maintain group cohesion in a chaotic environment. Although this has been solved for a single user, scaling to many users requires computational efficiency, safety on incomplete information, maintaining group cohesion, and rapid optimal advice.

This paper shows how to address this challenge and provide effective navigational advice by inferring behavioural traces of pedestrians in the chaotic environment, abstracting from individuals to groups of pedestrians, representing the groups in the social force model, and then finding an optimal navigation solution that maintains group cohesion and safety with statistical model checking. The technologies presented here are validated on an open data set of real world observations, showing the effectiveness and efficiency of this solution.

Validation of this work with walkers and users is planned within the ACANTO project. Another area of future work will be to track users between social groups and so exploit this for better behavioural inference and prediction. This could for example rely on the use of a user ID that is provided by the FriWalk. The parameters of various algorithms could also be automatically inferred during operation, rather than determined a priori, perhaps yielding more accurate and robust human motion models.

REFERENCES

[1] ACANTO project web site. http://www.ict-acanto.eu/, October 2017.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[3] M. Bennewitz, W. Burgard, and S. Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and autonomous systems*, 41(2):89–99, 2002.

[4] B. Boyer, K. Corre, A. Legay, and S. Sedwards. PLASMA-lab: A Flexible, Distributable Statistical Model Checking Library. In *Proceedings of QEST*, volume 8054 of *LNCS*, pages 160–164. Springer, 2013.

[5] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems, Revised Reprint:*. Other titles in applied mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2009.

[6] M. Čáp, P. Novák, A. Kleiner, and M. Selecký. Prioritized planning algorithms for trajectory coordination of multiple mobile robots. *IEEE Transactions on Automation Science and Engineering*, 12(3):835–849, 2015.

[7] M. Cáp, P. Novák, M. Selecký, J. Faigl, and J. Vokffnek. Asynchronous decentralized prioritized planning for coordination in multi-robot system. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3822–3829. IEEE, 2013.

[8] A. Colombo, D. Fontanelli, A. Legay, L. Palopoli, and S. Sedwards. Motion planning in crowds using statistical model checking to enhance the social force model. In *Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013, December 10-13, 2013, Firenze, Italy*, pages 3602–3608. IEEE, 2013.

[9] DALi project web site. http://www.ict-dali.eu/dali, September 2017.

[10] F. David. Map-based navigation in mobile robots: I. *A review of localization strategies: Cognitive Systems Research*, 4:243–282, 2003.

[11] V. R. Desaraju and J. P. How. Decentralized path planning for multi-agent teams with complex constraints. *Autonomous Robots*, 32(4):385–403, 2012.

[12] ETH Zï£¡rich BIWI walking pedestrians open dataset web site. http://www.vision.ee.ethz.ch/datasets/, October 2017.

[13] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo. When Helbing Meets Laumond: The Headed Social Force Model. In *Proc. IEEE Int. Conf. on Decision and Control (CDC)*, pages 3548–3553, Las Vegas, Nevada, US, Dec. 2016. IEEE.

[14] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo. Walking Ahead: The Headed Social Force Model. *PLOS ONE*, 12(1):1–23, 01 2017.

[15] T. Given-Wilson, A. Legay, and S. Sedwards. Information security, privacy, and trust in social robotic assistants for older adults. In T. Tryfonas, editor, *Human Aspects of Information Security, Privacy and Trust - 5th International Conference, HAS 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings*, volume 10292 of *Lecture Notes in Computer Science*, pages 90–109. Springer, 2017.

[16] D. Helbing, I. Farkas, P. Molnár, and T. Vicsek. Simulation of pedestrian crowds in normal and evacuation situations. In M. Schreckenberg and S. D. Sharma, editors, *Pedestrian and Evacuation Dynamics*. Springer, 2002.

[17] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, September 2000.

[18] D. Helbing, I. J. Farkas, and T. Vicsek. Freezing by heating in a driven mesoscopic system. *Phys. Rev. Lett.*, 84:1240–1243, 2000.

[19] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995.

[20] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet. Approximate probabilistic model checking. In *Verification, Model Checking, and Abstract Interpretation (VMCAI 2004)*, volume 2937 of *LNCS*, pages 73–84. Springer, 2004.

[21] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[22] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967.

[23] M. Mamei and F. Zambonelli. Physical deployment of digital pheromones through rfid technology. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 281–288. IEEE, 2005.

[24] H. Mo and L. Xu. Research of biogeography particle swarm optimization for robot path planning. *Neurocomputing*, 148:91–99, 2015.

[25] S. Nouyan, A. Campo, and M. Dorigo. Path formation in a robot swarm. *Swarm Intelligence*, 2(1):1–23, 2008.

[26] K. J. OâĂŹHara and T. R. Balch. Pervasive sensor-less networks for cooperative multi-robot tasks. In *Distributed Autonomous Robotic Systems 6*, pages 305–314. Springer, 2007.

[27] M. Okamoto. Some inequalities relating to the partial sum of binomial probabilities. *Annals of the Institute of Statistical Mathematics*, 10:29–35, 1959.

[28] PLASMA-lab SMC library web site. http://project.inria.fr/plasma-lab, October 2017.

[29] M. Vavrinec and M. Schaefer. Prioritized planning for road vehicles coordination. 2015.

[30] A. Wald. Sequential Tests of Statistical Hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[31] H. Younes and R. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *CAV*, volume 2404, pages 23–39. Springer, 2002.

[32] Y. Zhang, D.-w. Gong, and J.-h. Zhang. Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing*, 103:172–185, 2013.