2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DANCeRS: A Distributed Algorithm for Negotiating Consensus in Robot Swarms with Gaussian Belief Propagation

Aalok Patwardhan[1] and Andrew J. Davison[1]

***Abstract*— Robot swarms require cohesive collective behaviour to address diverse challenges, including shape formation and decision-making. Existing approaches often treat consensus in discrete and continuous decision spaces as distinct problems. We present DANCeRS, a unified, distributed algorithm leveraging Gaussian Belief Propagation (GBP) to achieve consensus in both domains. By representing a swarm as a factor graph our method ensures scalability and robustness in dynamic environments, relying on purely peer-to-peer message passing. We demonstrate the effectiveness of our general framework through two applications where agents in a swarm must achieve consensus on global behaviour whilst relying on local communication. In the first, robots must perform path planning and collision avoidance to create shape formations. In the second, we show how the same framework can be used by a group of robots to form a consensus over a set of discrete decisions. Experimental results highlight our method's scalability and efficiency compared to recent approaches to these problems making it a promising solution for multi-robot systems requiring distributed consensus. We encourage the reader to see the supplementary video demo.**

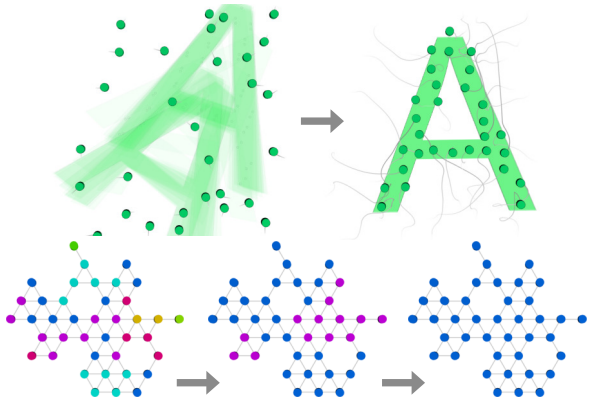


Fig. 1: (Top) Robots jointly plan paths to arrange into the 'A' shape, achieving consensus over its pose. (Bottom) Robots use the *same* algorithm to achieve consensus over a discrete set of decisions (colours) through local communication.

I. INTRODUCTION

In nature swarms such as bird murmurations or schools of fish consist of simple agents that communicate locally and exhibit collective behaviour, often requiring agreement on shared global parameters like velocity or orientation. Similarly, robot swarms must act cohesively, whether for task allocation, arrangement, or coordinated behaviours like exploration or aggregation [1]. This requires each robot to make individual decisions while aligning with the swarm's global objectives.

Centralised systems offer precise coordination by delegating all computation to a single external entity. However, such approaches lack scalability and are prone to single points of failure. In contrast, decentralised systems rely on distributed computation and local communication, enabling agents to share information and iteratively refine their decisions based on their neighbours' states. For discrete decision spaces, agents often share probability distributions over the available choices and update their beliefs until consensus is reached [2]. These methods are effective for swarms deciding between distinct behaviours like exploration [3], aggregation, or task allocation, known as *best-of-N* problems [4].

For continuous spaces, consensus is typically achieved by iteratively updating shared variables, such as positions or orientations, based on differences between neighbouring agents' states. This allows swarms to converge toward global parameters collaboratively, even with limited local communication. Such approaches have been used for shape formation [5], where robots must agree on the position and orientation of a shape for coordinated motion. These distributed methods provide scalability and robustness, enabling swarms to function in dynamic environments without centralised control.

Despite significant progress, the literature typically treats consensus over discrete decision spaces and continuous parameter spaces as separate problems. In this work, we propose a unified framework for achieving consensus in both domains. By modelling the swarm as a sparsely connected factor graph and using Gaussian Belief Propagation (GBP) for distributed inference, our method enables robots to reach consensus efficiently and scalably. This fully distributed approach ensures robustness to increasing swarm size and dynamic environments, addressing both discrete and continuous decision-making in a single cohesive framework.

We demonstrate our general framework in two example applications of joint path planning for shape formation and collective decision-making over a set of discrete options. Our contributions are:

- A unified, distributed framework for consensus in both discrete and continuous decision spaces.
- The extension of GBP path planning for non-holonomic unicycle dynamics in shape formation.
- A novel distributed shape formation method for discrete target assignment, demonstrated with scalability and efficiency in experiments.

[1]Aalok Patwardhan and Andrew J. Davison are with the Dyson Robotics Lab and the Department of Computing, Imperial College London [a.patwardhan21, a.davison]@imperial.ac.uk. This work was supported by Dyson Technology Ltd and EPSRC.

- The first application of GBP for discrete consensus in robot swarms.

II. RELATED WORK

Achieving global behaviour in robot swarms relies on decision-making frameworks that balance scalability and robustness. While centralised methods are limited by scalability and single points of failure, decentralised approaches enable collective behaviour through local interactions, whilst addressing challenges such as conflict resolution. The literature focuses on two key problems: consensus over (1) a set of discrete options, and (2) over continuous parameter spaces.

In discrete consensus problems, each agent in the swarm must select one option from a finite set based on local communication with its neighbours. This problem, often referred to as the "best-of-N" problem has been widely studied particularly for $N{=}2$ options where a swarm estimates the majority value in the environment [6], [7]. For larger N, probabilistic approaches involve agents maintaining and sharing a probability distribution over the set of options. In [8] agents update their internal beliefs based on local communication and knowledge of the local consensus group (agents sharing the same decision) whereas in [2] agents share decision IDs and levels of certainty of their beliefs based on probabilistic entropy. Other popular distributed approaches include voter methods where robots use information from their neighbours and choose a random decision [9], the majority decision [10] or the most frequent recently observed decision [11]. Another approach to distributed decision-making is opinion dynamics, where agents update their opinions through local interactions, allowing for agreement, disagreement, and clustering beyond traditional consensus models [12]. However, these models do not guarantee convergence to a common decision, which is often required in multi-agent coordination tasks.

Continuous consensus problems involve swarms negotiating beliefs over continuous global parameters, such as target locations in search-and-rescue or source-seeking, or the pose of a reference frame for shape formation. Distributed consensus algorithms typically rely on averaging neighbours' beliefs for tasks like formation control [13], obstacle avoidance [14], and saliency detection [15]. Mean-shift has been applied to shape formation [5], enabling robots to negotiate shape pose and form complex patterns via artificial potential fields, and later extended to distinct-point formations [16]. These methods align robot velocities with local density maxima, supporting collision avoidance and dynamic role assignment, but are limited to shapes with a single connected component. Other approaches, such as local task swapping [17], iteratively assign positions by exchanging roles based on local priorities, using hop-count strategies for collision-free movement on discretised grids. However, they assume prior consensus on a global reference frame. In contrast, we address shape formation and reference frame alignment jointly, removing the need for role-based task negotiation.

Despite being treated as two distinct problems in the literature, real-world tasks often require swarms to achieve consensus in both discrete and continuous decision spaces simultaneously. In our work, we propose a unified framework capable of addressing both types of consensus problems. Our approach builds on Gaussian Belief Propagation (GBP), a distributed algorithm for factor graph inference, which has demonstrated strong performance in dynamically changing graph topologies. GBP is asynchronous, fully distributed, and relies on peer-to-peer message passing between nodes in the graph. Previous works [3], [18] applied GBP to distributed multi-robot path planning and information aggregation but were limited to Euclidean spaces and holonomic motion models. Recently, GBP has also been used for consensus in 2D spaces [19], but without extending to Lie groups, restricting its ability to handle complex spaces.

In this work we introduce a general GBP-based framework for consensus over Lie groups, enabling robots to negotiate over more complex spaces beyond simple Euclidean variables. Unlike [3], where consensus was achieved through implicit measurement fusion, we formulate it as an explicit negotiation process. Additionally, we extend GBP to discrete decision spaces, broadening its applicability beyond continuous estimation. We also propose a novel shape formation algorithm, leveraging GBP for distributed coordination, and incorporate a non-holonomic motion model, making our approach more realistic than prior work for practical robotic systems.

III. BACKGROUND

A. Gaussian Belief Propagation (GBP) for Factor Graphs

Probabilistic Graphical Models (PGMs) are a powerful framework for representing sparse optimisation problems, where a set of variables $\mathbf{X}$ are connected by constraints. The joint probability function $p(\mathbf{X})$ can be factorised into factors f_s that encapsulate constraints over a subset of the variables $\mathbf{X}_s$:

$$p(\mathbf{X}) = \prod_s f_s(\mathbf{X}_s) \ . \tag{1}$$

When the factors and variables take on Gaussian distributions we can use Gaussian Belief Propagation (GBP) to perform marginal inference on the resulting factor graph. In GBP variables and factors are nodes which perform purely local node-to-node message passing and update their beliefs. At any time, the current marginal beliefs for the variables can be obtained. Following the convention from [20], we represent a Gaussian distribution $\mathcal{N}(\mathbf{X}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ in information form as $\mathcal{N}^{-1}(\mathbf{X}; \boldsymbol{\eta}, \boldsymbol{\Lambda})$, where $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$ are known as the precision matrix and information vector respectively. In GBP, variables in $\mathbf{X}_s$ are assumed to be Gaussian: each variable $\mathbf{x}_k$ has a belief $b(\mathbf{x}_k) = \mathcal{N}^{-1}(\mathbf{x}_k; \boldsymbol{\eta}_k, \boldsymbol{\Lambda}_k)$. A Gaussian factor $f_s(\mathbf{X}_s)$ can be any (non-linear) function that connects variables $\mathbf{X}_s$ of the form:

$$f_s(\mathbf{X}_s) \propto e^{-\frac{1}{2}\boldsymbol{r}^\top \boldsymbol{\Lambda}_s \boldsymbol{r}} \ , \tag{2}$$

where the residual $\boldsymbol{r} = \boldsymbol{z}_s - \boldsymbol{h}_s(\mathbf{X}_s)$, $\boldsymbol{h}_s(\mathbf{X}_s)$ represents the linearised measurement function or constraint that the factor represents, $\boldsymbol{z}_s$ is its observed value and $\boldsymbol{\Lambda}_s$ is the precision of the constraint.

One iteration of GBP consists of three steps involving node-to-node message passing which we outline briefly here; see [20] for a more in-depth analysis and derivation.

1) Factor to Variable Message: A message from a factor f_j to one of its connected variables $\mathbf{x}_k$ is computed by marginalizing the product of the factor's potential (the exponent in Equation 2) and all incoming variable to factor messages, excluding the message from $\mathbf{x}_k$.

2) Variable Belief Update: A variable $\mathbf{x}_k$ updates its belief by taking the product of all incoming messages from its connected factors.

3) Variable to Factor Message: A message from a variable $\mathbf{x}_k$ to a connected factor f_j is the product of all incoming factor-to-variable messages except from f_j.

B. GBP using Lie Theory

Gaussian Belief Propagation (GBP) is commonly used in Euclidean spaces, where variables are represented as vectors. However, many robotics problems require optimisation over Lie groups, which provide a natural way to represent transformations, orientations, and motions. Lie groups are continuous mathematical structures that generalise Euclidean spaces while preserving properties like smoothness and group operations. They are widely used in robotics to represent states such as positions, rotations, and velocities in a way that respects the underlying geometry.

In consensus problems, robots must agree on shared parameters such as position, orientation, or categorical decisions. While averaging works well for simple Euclidean quantities, it fails for rotations and transformations e.g. the Euclidean average of 1° and 359° is 180°. Lie theory allows optimisation to be performed on the correct mathematical space, ensuring accurate consensus.

GBP operates by passing Gaussian-distributed messages between variables in a factor graph. In Euclidean settings, GBP messages consist of a mean vector and precision matrix, but for Lie groups, they must be transformed into the tangent space of the current belief, updated, and mapped back using exponential Exp(·) and logarithmic Log(·) maps. This ensures that information is propagated correctly without distorting the underlying geometry. For a detailed derivation and further insights, see [21], [22]. Our method provides a general framework for optimisation on Lie groups, enabling applications in trajectory planning, state estimation, and control. In this work, we demonstrate its use for multi-robot consensus over poses and discrete decisions.

IV. METHOD

We consider a swarm of N robots moving in $\mathbb{R}^2$ each having a communications radius r_C. At any time $t \geq 0$ they form a sparsely connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{\mathcal{V}_i : i \in [0, ..., N-1]\}$ is the set of robots, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges denoting inter-robot connections. An edge $\mathcal{E}_{ij}$ between robot i and j exists at time t if $\left\|\boldsymbol{x}^i - \boldsymbol{x}^j\right\| < r_C$, where $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$ are the 2D positions of robots i and j respectively.

The robots must solve multifaceted problems; they must perform optimisation in path planning and collision avoidance, but also form consensus over some shared global information. Similarly to the approach from [3], each robot holds a two-layered factor graph stack as shown in Figure 2 where each layer of the stack represents a particular aspect of the optimisation problem: the Planning $\mathcal{P}$ and Global Consensus $\mathcal{G}$ layers.

As ours is a purely distributed algorithm, we take inspiration from [21]; each robot maintains a "webpage" of information containing the outgoing messages to the factors and variables in the GBP stacks of each of its connected neighbours. Each robot follows Algorithm 1.

The parametrisation of variables in the Global Consensus layer depends on the specific application to be considered. To demonstrate our general framework which works for many of the common Lie Groups, we perform two types of experiments where the robots must reach consensus over some shared global information.

In the first experiment, we choose shape formation consensus as a proxy problem where the $\mathrm{SE}(2)$ group can be used to represent the position and orientation of a target formation shape. In the second, we examine how the same framework can be used for performing negotiation over the continuous $\mathbb{R}^M$ group, and can be used to perform decision-making and negotiation over a set of discrete options.

We now describe the general form of the two factor graph layers as depicted in Figure 2. In general we describe a factor f with its measurement function $\boldsymbol{h}$ and its strength given by vector $\boldsymbol{\sigma}$ such that the measurement precision $\boldsymbol{\Lambda} = diag(\boldsymbol{\sigma}^{-2})$ is a diagonal matrix. A GBP variable is denoted as ${}^{\mathcal{L}}\mathbf{x}_m^R$, where R is the robot index, $\mathcal{L}$ is the GBP Stack layer, and m is the index of the variable in that layer.

A. Global Consensus Layer $\mathcal{G}$

In this layer robots may share information about their interpretation of the global parameters and negotiate to collectively form a consensus.

Let χ represent the global parameter that all robots must agree upon. For example, this may represent the index of a shared behaviour to be exhibited (from a list of different behaviours e.g. [*explore, regroup, return, ...*]), or alternatively a common reference vector to be agreed upon by the robots, such as a common origin. Each robot holds variables ${}^{\mathcal{G}}\mathbf{x} \in \mathcal{M}$ representing the robot's interpretation of the global parameters. In our generalised framework $\mathcal{M}$ may be any one of the common Lie groups such as $\mathbb{R}^M, \mathrm{SO}(2), \mathrm{SO}(3), \mathrm{SE}(2), \mathrm{SE}(3)$.

1) Consensus Domain: A GBP variable ${}^{\mathcal{G}}\mathbf{x}^i$ in this layer represents robot i's belief about the global parameter χ, and consensus is achieved over the continuous domain $\mathcal{M}$.

When applying our method to problems involving consensus over a discrete set of options, we choose $\mathcal{M} = \mathbb{R}^1$ to represent the continuous form of the discrete decision space. We define a quantisation function $\gamma : [0, 1) \rightarrow$

$\{0, 1, \ldots, N_D - 1\}$ and its inverse γ^{-1} as

$$\gamma(x) = \lfloor N_D \cdot x \rfloor \,; \quad \gamma^{-1}(k) = \frac{k}{N_D} \,, \text{ for } k = 0, \ldots, N_D - 1 \tag{3}$$

respectively where $N_D \in \mathbb{Z}_{>0}$ is the total number of discrete intervals. Transforming the discrete decision space into a continuous space allows the use of GBP for iterative negotiation, enabling the swarm to converge onto a shared common decision. It should be noted that the quantisation function is used simply to extract the robot's discrete decision, and is not part of the consensus algorithm.

2) Prior Factor: Robot i has a prior belief about the global parameter represented by a factor of strength $\boldsymbol{\sigma}_p$:

$$f_p : \boldsymbol{h}_p({}^{\mathcal{G}}\mathbf{x}^i) = {}^{\mathcal{G}}\mathbf{x}^i \ \ominus \ {}^{\mathcal{G}}\boldsymbol{x}^{i,0} \,, \tag{4}$$

where ${}^{\mathcal{G}}\boldsymbol{x}^{i,0}$ is the initial belief of the formation parameters, and $\ominus$ is the right–minus action on the manifold.

3) Inter-Robot Consensus Factor: When two robots are within communication range of each other, inter-robot factors are created from each variable in this layer. The factors have strength $\boldsymbol{\sigma}_c$ and take the form

$$f_c : \boldsymbol{h}_c({}^{\mathcal{G}}\mathbf{x}^i, {}^{\mathcal{G}}\mathbf{x}^j) = {}^{\mathcal{G}}\mathbf{x}^i \ominus {}^{\mathcal{G}}\mathbf{x}^j = \mathrm{Log}\left(({}^{\mathcal{G}}\mathbf{x}^j)^{-1} \circ {}^{\mathcal{G}}\mathbf{x}^i\right) \,, \tag{5}$$

where $\circ$ is the Lie Group composition operator.

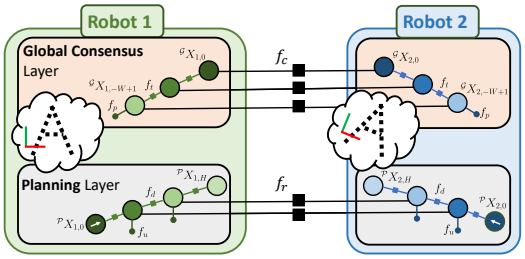


Fig. 2: Factor graph stack of each robot. Local GBP iterations enable global consensus between the robots, visualised here as the formation parameters for the letter A.

4) Temporal Factors and Variables: The resulting factor graph is dynamic as robots may leave and rejoin a local group at any time. When robots leave a group, they delete their existing inter-robot factors – as GBP variables are memoryless, robots would forget the locally converged belief of the global parameter. Hence, robots maintain a sliding window of W variables ${}^{\mathcal{G}}\mathbf{x}^i_{-k} : k \in [0, W-1]$ in this layer, which are connected with temporal factors $f_t({}^{\mathcal{G}}\mathbf{x}^i_{-k+1}, {}^{\mathcal{G}}\mathbf{x}^i_{-k})$ having the same form as the inter-robot factor in Equation 5 with a strength $\boldsymbol{\sigma_t}$. The sliding window is updated every T_S time steps. The oldest variable ${}^{\mathcal{G}}\mathbf{x}^i_{-W+1}$ along with its connected factors is deleted. A new prior factor f_p is generated at the now oldest variable ${}^{\mathcal{G}}\mathbf{x}^i_{-W+2}$ and initialised with the message from factor $f_t({}^{\mathcal{G}}\mathbf{x}^i_{-W+1}, {}^{\mathcal{G}}\mathbf{x}^i_{-W+2})$. This prior factor represents the marginalised information from the deleted portion of the factor graph.

The resulting effect is that a robot disconnecting from a local group maintains its previous belief mean, although its covariance weakens over the sliding window.

B. Path Planning Layer $\mathcal{P}$

Optimisation in this layer involves robots planning an efficient path in a short forward time window of K time steps, and creating inter-robot factors between themselves and their neighbours for collision avoidance.

A noise-on-acceleration model is used for the robot dynamics such that the robot states are parametrised as

$${}^{\mathcal{P}}\mathbf{x} = [\boldsymbol{p}^\top, \dot{\boldsymbol{p}}^\top]^\top = [x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}]^\top \,. \tag{6}$$

We build upon the work presented in [18] where robots with holonomic dynamics models planned paths towards fixed goals. These paths consisted of the robots' planned states ${}^{\mathcal{P}}\mathbf{x}_k$ for $k \in [0, K-1]$ which were the optimisation variables, connected to each other through dynamics factors f_d representing a constant velocity model constraint. In this work however we propose a new unicycle model factor f_u, enabling modelling of non-holonomic motion, and improvements over the previous work, which we detail here.

1) Unicycle Model Factor: This factor with strength σ_u is added to each variable in the layer and takes the form:

$$f_u : \boldsymbol{h}_u({}^{\mathcal{P}}\mathbf{x}_k) = \dot{x}\cos(\theta) - \dot{y}\sin(\theta) \,. \tag{7}$$

As factors represent cost functions, optimisation drives the value of the measurement function of this factor $\boldsymbol{h}_u({}^{\mathcal{P}}\mathbf{x}_k)$ to 0. This encourages the robot velocity at any time k to be in the direction of the robot's heading.

2) Collision Avoidance Factor: We also introduce a new inter-robot collision avoidance factor between the variables of robot i and j which has strength σ_r and takes the form:

$$f_r : \boldsymbol{h}_r({}^{\mathcal{P}}\mathbf{x}_{k,i}, {}^{\mathcal{P}}\mathbf{x}_{k,j}) = \exp\left\{-\frac{|\boldsymbol{x}_{k,i} - \boldsymbol{x}_{k,j}|}{d_{min}}\right\} \,, \tag{8}$$

where d is the distance between the two robot planned paths at time step k, and d_{min} is a minimal separation distance between the robots. This factor has a smoother measurement function compared to the equivalent factor in [18] and so a more well-defined smoother Jacobian.

3) Horizon Update: As per Algorithm 1, at every time step after N_I iterations of GBP the current and horizon variables of the planned path are propagated forward in time. Unlike previous work, we move the horizon state $\boldsymbol{p}_{K-1}$ towards an application-specific goal location $\boldsymbol{g}$ which can be time-varying. First we create the goal vector $\boldsymbol{d}$ as the straight-line vector from the horizon state's position $[\ \boldsymbol{p}_{K-1}[x],\ \boldsymbol{p}_{K-1}[y]\ ]^\top$:

$$\boldsymbol{d} = [\ \boldsymbol{g}[x] - \boldsymbol{p}_{K-1}[x],\ \boldsymbol{g}[y] - \boldsymbol{p}_{K-1}[y]\ ]^\top \,. \tag{9}$$

Secondly, the goal vector is used to update the velocity of the horizon state $\dot{\boldsymbol{p}}_{K-1}$, capping the linear and angular velocities at v_{max}, and ω_{max}:

$$\dot{\boldsymbol{p}}_{K-1} = \begin{bmatrix} \min\left(v_{\max}, \frac{\|\boldsymbol{d}\|}{\Delta t}\right) \cdot \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} \\ \min\left(\omega_{\max}, \frac{|\angle \boldsymbol{d} - \boldsymbol{p}_{K-1}[\theta]|}{\Delta t}\right) \cdot \mathrm{sign}(\angle \boldsymbol{d} - \boldsymbol{p}_{K-1}[\theta]) \end{bmatrix} \,. \tag{10}$$

Algorithm 1 For each robot i

Initialise GBP stack layers $\mathcal{P}, \mathcal{G}$
while $t < T_{max}$ **do**
  Let $\mathcal{N}(i) = \{j \mid ||\boldsymbol{p}_0^i - \boldsymbol{p}_0^j|| < r_C\}$ *be the set of robots within the communication radius of* i.
  Let $\mathcal{C}(i)$ *be the set of robots connected to* i.
  for Newly observed robot $j \in \mathcal{N}(i) \backslash \mathcal{C}(i)$ **do**
    Create inter-robot factors f_c, f_r.
  for Out-of-range robot $j \in \mathcal{C}(i) \backslash \mathcal{N}(i)$ **do**
    Delete inter-robot factors f_c, f_r.
  for layer $\mathcal{L}$ in GBP Stack **do**
    Broadcast inter-robot messages.
    Perform N_I iterations of GBP.
  Update ${}^{\mathcal{P}}\mathbf{x}_0$ and ${}^{\mathcal{P}}\mathbf{x}_{K-1}$ by Δt.
  Create new variables ${}^{\mathcal{G}}\mathbf{x}_k$ with temporal factors f_t.

Finally this new velocity is used to update the position of the horizon state $\boldsymbol{p}_{K-1}$ with a simple dynamics model:

$$\boldsymbol{p}_{K-1} \leftarrow \boldsymbol{p}_{K-1} + \dot{\boldsymbol{p}}_{K-1}\Delta t \ . \tag{11}$$

If the specific application requires the robots to randomly explore the environment, a new $\boldsymbol{g}$ is selected when the robot's horizon state is sufficiently near it. Alternatively if the swarm is to create a shape formation, $\boldsymbol{g}$ can be selected using a nearest-neighbour search over potential locations using a novel distance-occupancy based heuristic we propose in section V-A.

V. APPLICATIONS AND SIMULATIONS

A. Path Planning and Consensus for Shape Formation

A swarm of robots with limited communication radius r_C must plan paths to arrange themselves into a formation. Each robot knows the target shape but must negotiate the formation parameters (position and orientation) with others. The shape is defined as a set of points in $\mathbb{R}^2$ with minimum spacing r_S. Following Algorithm 2 robots select the nearest point as the goal $\boldsymbol{g}$ using a heuristic based on distance and local occupancy.

The formation consists of $N_F = N_R$ points $q_n \in \mathbb{R}^2$ for $n \in N_F$, known by all robots in a canonical frame. Robots use their belief of the formation parameters ${}^{\mathcal{G}}\mathbf{x}^i = \mathrm{Exp}\left([x^i,\ y^i,\ \theta^i]^\top\right)$ to transform points from canonical to global frames via the group action $\boldsymbol{q}' = {}^{\mathcal{G}}\mathbf{x}^i \circ \boldsymbol{q}$. Although we use $\mathrm{SE}(2)$ to represent position and orientation, other representations such as $\langle \mathbb{R}^2, \mathrm{SO}(2) \rangle$ are also possible [22].

1) Occupancy Weighting: Robots must select points unoccupied by neighbours, which is difficult under limited communication. For each robot i the set of formation points (FPs) Q^i is augmented with 'occupancy weighting' (OW) values τ_n, yielding

$$\widetilde{Q^i} = \left\{ [q_n^{i\ \top}\ \tau_n^i]^\top \ \forall\ q_n^i \in Q^i \right\} \ . \tag{12}$$

Each robot stores $\widetilde{Q^i}$ in a KD-Tree for efficient nearest-neighbour search. The OW dimension encodes how recently a point was seen as occupied. If any neighbouring robots are near (within a range r_N) to an FP q_n^i, its OW is set at an arbitrarily high value of $\tau_n^i = \tau_0$. The OW values of the remaining FPs within a distance r_C of the robot are set at 0 as they appear unoccupied. For all other FPs, their OW values are decremented by 1, indicating that they may still be occupied, as shown in Figure 3.

2) Goal Selection: A robot i transforms its position $\boldsymbol{p}_0^i$ into the canonical frame, augmenting it with a 0 in the OW dimension. A nearest-neighbour search across $\widetilde{Q^i}$ then favours FPs with low occupancy. As seen in the supplementary video and in Figure 4, without OW, robots can become stuck oscillating between FPs when communication is lost with occupying neighbours.

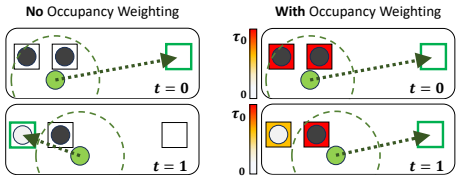


Fig. 3: Occupancy Weighting (OW): at $t = 0$ the green robot selects the nearest formation point *unoccupied* by its neighbours (gray) to move towards. At $t = 1$ the left-most neighbour is out of comms. range and the point it is on seems ideal. With OW, the right-most point remains optimal.

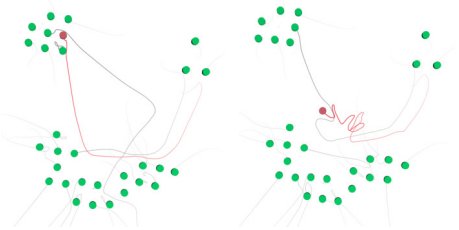


Fig. 4: Left: occupancy weighting (OW) encourages shape exploration. Right: without OW the final (red) robot oscillates, because it 'forgets' occupancy information about a point once out of range of the occupying robots.

B. Experiments: Continuous Consensus – Shape Formation

Robots of radius 1 m are initialised with random positions and headings $\boldsymbol{p}_i$ in a 100×100 m^2 environment. They follow a non-holonomic unicycle dynamics model, and plan paths with a time window of $T_H = 1.5$ s towards randomly generated goals with a maximum velocity $v_{max} = 2$ m/s and a turning radius $r_T = 2$ m. Additionally we set $\sigma_u = 0.001$,

Algorithm 2 For each robot i

for $j \in \mathcal{N}(i)$ **do**
  for $\widetilde{q}^i \in \widetilde{Q^i}$ **do**
    if $\left\| [\mathbf{I}\ \mathbf{0}]\, \widetilde{q}^i - ({}^{\mathcal{G}}\mathbf{x}^i)^{-1} \circ \boldsymbol{p}_0^j \right\| < r_N$ **then**
      $\widetilde{q}^i(\tau) \leftarrow \tau_0$
for $\widetilde{q}^i \in \widetilde{Q^i}$ **do**
  if $\left\| [\mathbf{I}\ \mathbf{0}]\, \widetilde{q}^i - ({}^{\mathcal{G}}\mathbf{x}^i)^{-1} \circ \boldsymbol{p}_0^i \right\| < r_C$ **then**
    $\widetilde{q}^i(\tau) \leftarrow 0$
  else
    $\widetilde{q}^i(\tau) \leftarrow \max\left(0,\ \widetilde{q}^i(\tau) - 1\right)$
  $\mathbf{g}_i = {}^{\mathcal{G}}\mathbf{x}^i \circ [\mathbf{I}\ \mathbf{0}] \left(\underset{\widetilde{q}^i \in \widetilde{Q^i}}{\arg\min} \left\| \widetilde{q}^i - [({}^{\mathcal{G}}\mathbf{x}^i)^{-1} \circ \boldsymbol{p}_0^i \quad 0]^\top \right\| \right)$

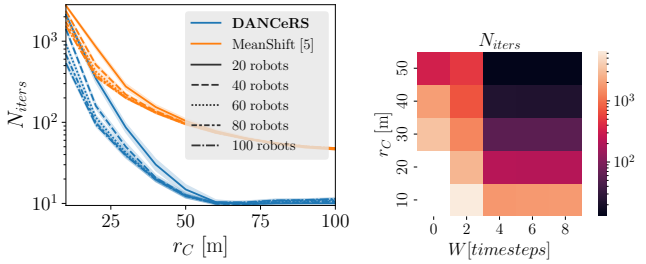
$\sigma_d = 0.1$, $\sigma_r = 0.01$. In all of our experiments in this work we perform *one* iteration of inter-robot message passing per time step only, and set $N_I = 2$. For all experiments in this work we present results over 50 trials and a range of letter formations with $d_{min} = r_N = 2$ m, $r_S = 4$ m, and $\tau_0 = 10^3$.

In the Global Consensus layer, each robot holds a sliding window of $W = 3$ variables, updated every time step ($T_S = 1$). The first of these variables is initialised with a prior factor with observation $\boldsymbol{p}_0^i$. The factor strengths are set with $\boldsymbol{\sigma_p} = [10, 10, \pi]$, $\boldsymbol{\sigma_s} = 0.1\boldsymbol{\sigma_p}$, and $\boldsymbol{\sigma_r} = 0.01\boldsymbol{\sigma_p}$.

1) Convergence on Formation Parameters: We vary the radius of communication r_C and the number of robots N_R and measure the number of message passing iterations taken for the swarm to reach a consensus on the formation parameters. We define the convergence criteria to be when the mean inter-robot deviation in the robots' beliefs is less than 0.1m in position, and less than 0.01 rad in heading.

We compare against the distributed consensus algorithm used in [5], where agents effectively average their interpretations of formation position and orientation in their local groups. We use the best performing values ($c_1 = c_2 = 1.6$, $\alpha = 0.8$) from the work and consider the scenario of a static formation (stationary with respect to time).

Figure 5a shows that we achieve an order of magnitude improvement over the baseline. As r_C increases robots are seen to converge to the global formation parameters in fewer iterations as they are able to exchange information with more robots. This effect is also seen as the number of robots N_R, and therefore their density in the environment increases.



(a) (b)

Fig. 5: (a): As the communications radius r_C and number of robots N_R increase, it takes fewer iterations for the swarm to converge onto a consensus on the formation parameters. (b): As sliding window size W and r_C increase, the number of iterations to convergence decreases, shown for $N_R = 30$.

2) Effect of Sliding Window on Convergence: We investigate the effect of the length of the sliding window W of variables in the Global Consensus layer on the number of iterations taken until convergence. We consider the case of $N_R = 30$ and vary r_C. Figure 5b shows that increasing W results in faster convergence of the formation parameters within the swarm. This is more prominent as r_C increases.

When the radius of communication is small, robots forming a local group temporarily negotiate on their beliefs of the formation parameters. However as the variables in GBP are memory-less, when robots leave their local group (altering the factor graph topology) they would lose the previously negotiated belief. Instead using a sliding window of variables, robots are able to maintain their most recent belief due to the new marginalisation factor f_p attached to the now oldest variable. Figure 6 shows that the effect of the sliding window is to maintain the belief of the oldest variable but increase its covariance. With a longer window a robot's belief becomes more susceptible to change when it encounters a new clique of robots with a differing belief.

3) Shape Formation: We present qualitative experiments on robots jointly planning paths to form a pre-defined shape whilst achieving consensus on its position and orientation. The stopping criteria for the experiments is when the distance from each point within the formation to its nearest robot is less than a threshold r_R m. We perform experiments over a range of shapes including those in Figures 1, 6, 7, and the supplementary video.

Compared to the target-free, mean-shift based shape formation method in [5] which is applicable only to shapes consisting of one connected component, in our method robots explore the shape when they observe that all formation points nearby are occupied by other robots. We can thus handle disjoint shapes comprising sparse regions such as the Smiley Face formation, which mean-shift based methods cannot.

C. Consensus over a Discrete Decision Space

We now show how the same framework used for the *continuous* consensus problem can be used for reaching consensus over a *discrete* decision space. This problem can be thought of as a variation on the classical *Best-of-N* problem, where a swarm of agents with limited sensing and communication capabilities must reach agreement on a single decision from a discrete set of N_D possible options. This set may represent specific target locations for the swarm to visit, or simply various behaviours for the swarm to exhibit (e.g. 'aggregation', 'exploration', 'formation'). The critical objective is for all agents to agree on the same choice, as consensus within the swarm is essential for coherent collective action.

1) Global Consensus Layer: We consider that robots only perform optimisation over the Global Consensus layer $\mathcal{G}$ and do not perform path planning. Robots each hold one variable in this layer, randomly initialised with a decision $d^i \in [0, N_D - 1]$. For a robot i, a unary prior factor f_p with observation $\gamma^{-1}(d^i)$ and strength given by σ_p is connected to the variable ${}^{\mathcal{G}}\mathbf{x}^i$.

D. Experiments: Discrete Consensus

To validate our algorithm we compare against two recent distributed solutions to the discrete consensus problem: the Entropy-based consensus algorithm (ECA) [2] and the Probabilistic decision-making consensus algorithm (PCA) [8]. In ECA, robots maintain a probability distribution over the N_D options and exchange with neighbours both their strongest choice and the entropy of their distribution, representing certainty about the decision. In PCA, robots exchange their most preferred decision along with a list of neighbours

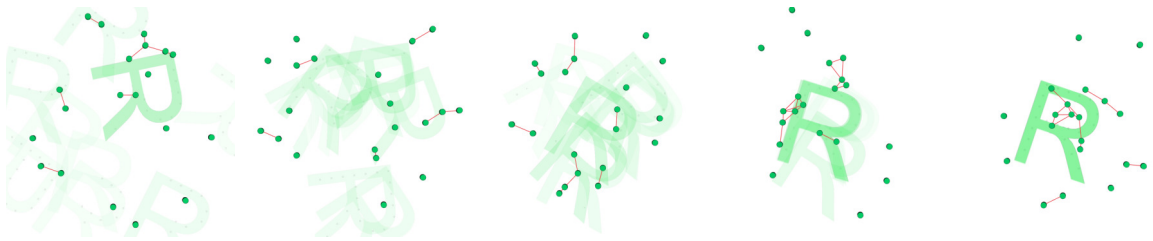


Fig. 6: Robots with a very small comms. radius r_C move randomly across the environment creating a dynamic graph (red lines show connectivity). Due to the sliding window of Global Consensus variables they are able to gradually form a consensus on the formation parameters (visualised with the shaded letter R).

Fig. 7: Qualitative examples of the shapes our method can enable swarms to create. Robot paths are shown in gray. Joint path planning and shape formation is possible even for shapes with multiple disconnected components ('!' and 'wifi').

exhibiting the same decision (their 'local consensus group'), and then update their states based on the received information. We use these algorithms as baselines since they outperform classical consensus methods such as majority rule [10] and k-unamity [11], and have demonstrated scalability. For completeness, we also include comparisons against [5], which, like our approach, addresses consensus in a continuous decision space.

It should be noted that PCA [8] assumes fixed connectivity in the robot network. It also requires that when a robot joins or leaves a local consensus group, the next iteration only proceeds after this update has reached all robots. In real-world networks with dynamic connectivity, this assumption is often unrealistic and can increase convergence time.

Following the experimental setup in [8], robots are placed in a random triangular grid (Figure 1, bottom) such that the distance between any pair of robots is at least 5 m. Robots are initialised with random preferred decisions d^i, and the communication radius r_C is varied in increments of 6 m. We define a swarm as having converged when all agents exhibit the same discrete decision.

1) Discrete Convergence: We measure the number of message passing iterations N_{iters} until convergence as we vary N_R and the communications radius r_C. Figure 8 shows that as r_C increases it takes fewer iterations for the swarm to converge onto a common decision. For small r_C it takes longer to converge as the swarm size increases.

We plot only the trials that converged within 1000 time steps. Notably when $r_C = 6$ m the ECA method failed to converge, causing many local high-certainty decision groups. At higher r_C, our method demonstrated its scalability, taking a constant number of iterations to converge as N_R increased.

For the PCA algorithm we followed their assumption that changes to local decision groups are propagated throughout the swarm instantaneously, although in real-world experiments this would vastly increase N_{iters} with N_R as the amount of information shared between robots increases over time as the size of the dominant decision group increases.

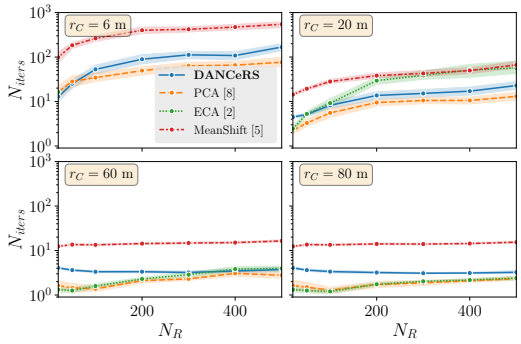


Fig. 8: Number of iterations N_{iters} taken for the swarm to converge to the same discrete decision as communication radius r_C and number of robots N_R vary.

2) Effect of sigma parameter sweep: We perform an ablation over the strength σ_c of the inter-robot consensus factors f_c and the number of discrete decisions N_D. In GBP the 'strength' of a factor reflects the extent to which a variable's belief may change during optimisation. We hypothesise that an ideal value of σ_c is $\frac{0.5}{N_D}$, as the γ partitions the continuous domain $[0, 1]$ into N_D discrete bins.

In Figure 9 the blue graphs show how the choice of σ_c and N_D affect the time to convergence, with the proportion of successful trials (within 1000 time steps) shown in red. Larger values of σ_c relate to weaker inter-robot consensus factors, leading to fewer successful trials. The vertical blue dotted lines at $\sigma = \frac{0.5}{N_D}$ indicate a favourable upper bound for the factor strength, where there were very few trials that did not converge. Smaller values of σ_c would result in stronger beliefs after convergence, and would not be suitable for problems where the global consensus parameter is time-

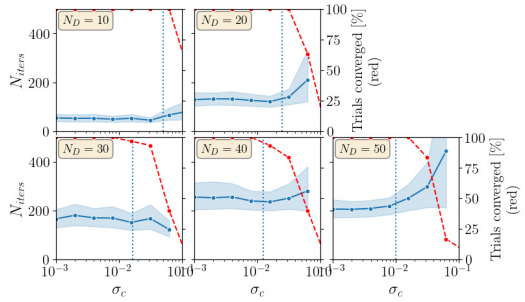


Fig. 9: Effect of consensus factor strength σ_c and number of decisions N_D on convergence (blue), and % of total trials that converged (red). Vertical dashed lines show $\sigma_c = \frac{0.5}{N_D}$.

varying; this is a promising avenue for future work.

3) Informed Robots: A proportion of the swarm may be 'informed' agents, or may have access to salient information about the decision to exhibit. Or, it may be desirable to control the large swarm through a small number of agents. We vary the proportion ζ of such 'seed' robots, which all exhibit the same decision and measure the proportion of the swarm that converges to the same decision within 1000 time steps. We consider the difficult case of $N_R = 500$ and $r_C = 6$ m: each robot in the triangular grid only communicates with its immediate neighbours.

For our method, we use a strong prior factor on the robot's Global Consensus layer variable with $\sigma_p = 10^{-10}$. Table I shows that our method resulted in convergence of the swarm to the seed robot decision throughout the range of seed robot proportions considered, from $\zeta = 0.002$ to $\zeta = 0.1$ (corresponding to 1 and 50 seed robots respectively).

TABLE I: % Trials converged as proportion of seed robots ζ varies for $N_R = 500$ and $r_C = 6$ m.

ζ	0.002	0.01	0.02	0.05	0.10	0.15	0.20
ECA [2]	0	0	0	0	0	0	0
PCA [8]	9	18	31	94	100	100	100
DANCeRS	80	100	100	100	100	100	100

VI. CONCLUSIONS

We presented DANCeRS, a distributed algorithm for consensus in robot swarms using Gaussian Belief Propagation (GBP). By modelling coordination as a factor graph, we enable scalable, decentralised decision-making in both continuous and discrete domains with only peer-to-peer communication. Its flexibility was demonstrated through two applications: shape formation with path planning, and collective decision-making over discrete options. The lightweight, fully distributed nature of GBP makes the method suitable for low-power devices: each robot optimises only its local portion of the factor graph. Limitations include the need to store all formation points, and the assumption of fixed options in discrete consensus. Despite these, DANCeRS offers a scalable, distributed solution for real-world multi-robot coordination, enabling negotiation of both continuous and discrete decisions in dynamic environments. **We encourage the reader to view our supplementary video**.

REFERENCES

[1] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7:1–41, 03 2013.

[2] Chuanqi Zheng and Kiju Lee. Consensus decision-making in artificial swarms via entropy-based local negotiation and preference updating. *Swarm Intelligence*, 17:1–21, 05 2023.

[3] Aalok Patwardhan and Andrew J. Davison. A distributed multi-robot framework for exploration, information acquisition and consensus. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12062–12068, 2024.

[4] Gabriele Valentini, Eliseo Ferrante, and Marco Dorigo. The best-of-n problem in robot swarms: formalization, state of the art, and novel perspectives. *Frontiers in Robotics and Artificial Intelligence*, 4, 2017.

[5] Guibin Sun, Rui Zhou, Zhao Ma, Yongqi Li, Roderich Groß, Zhang Chen, and Shiyu Zhao. Mean-shift exploration in shape assembly of robot swarms. *Nature Communications*, 14, 12 2023.

[6] Qihao Shan and Sanaz Mostaghim. Discrete collective estimation in swarm robotics with distributed bayesian belief sharing. *Swarm Intelligence*, 15(4):377–402, Dec 2021.

[7] Thiemen Siemensma, Darren Chiu, Sneha Ramshanker, Radhika Nagpal, and Bahar Haghighat. Collective bayesian decision-making in a swarm of miniaturized robots for surface inspection. In *Swarm Intelligence: 14th International Conference, ANTS 2024, Proceedings*, page 57–70. Springer-Verlag, 2024.

[8] Yang Liu and Kiju Lee. Probabilistic consensus decision making algorithm for artificial swarm of primitive robots. *SN Applied Sciences*, 2, 01 2020.

[9] Tim Rogers and Thilo Gross. Consensus time and conformity in the adaptive voter model. *Phys. Rev. E*, 88:030102, Sep 2013.

[10] Jan Wessnitzer and Chris Melhuish. Collective decision-making and behaviour transitions in distributed ad hoc wireless networks of mobile robots: Target-hunting. In *Advances in Artificial Life*, pages 893–902. Springer Berlin Heidelberg, 2003.

[11] Alexander Scheidler, Arne Brutschy, Eliseo Ferrante, and Marco Dorigo. The k-unanimity rule for self-organized decision making in swarms of robots. *IEEE Transactions on Cybernetics*, 46:1175, 2016.

[12] Anastasia Bizyaeva, Alessio Franci, and Naomi Ehrich Leonard. Nonlinear opinion dynamics with tunable sensitivity. *IEEE Transactions on Automatic Control*, 68(3):1415–1430, 2023.

[13] Andrey Savkin, Chao Wang, Ahmad Baranzadeh, Zhiyu Xi, and Hung Nguyen. Distributed formation building algorithms for groups of wheeled mobile robots. *Robotics and Autonomous Systems*, 75:463–474, 2016.

[14] Ryandika Afdila, Fahmi Fahmi, and Arman Sani. Distributed formation control for groups of mobile robots using consensus algorithm. *Bulletin of Electrical Engineering and Informatics*, 12:2095–2104, 2023.

[15] Merihan Alhafnawi, Sabine Hauert, and Paul O'Dowd. Self-organised saliency detection and representation in robot swarms. *IEEE Robotics and Automation Letters*, 6(2):1487–1494, 2021.

[16] Yunjie Zhang, Rui Zhou, Xing Li, and Guibin Sun. Mean-shift shape formation of multi-robot systems without target assignment. *IEEE Robotics and Automation Letters*, 9(2):1772–1779, 2024.

[17] Hanlin Wang and Michael Rubenstein. Shape formation in homogeneous swarms using local task swapping. *IEEE Transactions on Robotics*, 36(3):597–612, 2020.

[18] Aalok Patwardhan, Riku Murai, and Andrew J. Davison. Distributing collaborative multi-robot planning with gaussian belief propagation. *IEEE Robotics and Automation Letters*, 8(2):552–559, 2023.

[19] Simon Jones and Sabine Hauert. Distributed spatial awareness for robot swarms. In *17th International Symposium on Distributed Autonomous Robotic Systems*. Springer, Cham, 2024.

[20] Andrew J. Davison and Joseph Ortiz. Futuremapping 2: Gaussian belief propagation for spatial ai. *arXiv:1910.14139*, 2019.

[21] Riku Murai, Joseph Ortiz, Sajad Saeedi, Paul H. J. Kelly, and Andrew J. Davison. A robot web for distributed many-device localization. *IEEE Transactions on Robotics*, 40:121–138, 2024.

[22] Jérémie Deray and Joan Solà. Manif: A micro lie theory library for state estimation in robotics applications. *Journal of Open Source Software*, 5:1371, 02 2020.