

A Distributed Multi-Robot Framework for Exploration, Information Acquisition and Consensus

Aalok Patwardhan¹ and Andrew J. Davison¹

Abstract—The distributed coordination of robot teams performing complex tasks is challenging to formulate. The different aspects of a complete task such as local planning for obstacle avoidance, global goal coordination and collaborative mapping are often solved separately, when clearly each of these should influence the others for the most efficient behaviour. In this paper we use the example application of distributed information acquisition as a robot team explores a large space to show that we can formulate the whole problem as a single factor graph with multiple connected layers representing each aspect. We use Gaussian Belief Propagation (GBP) as the inference mechanism, which permits parallel, on-demand or asynchronous computation for efficiency when different aspects are more or less important. This is the first time that a distributed GBP multi-robot solver has been proven to enable intelligent collaborative behaviour rather than just guiding robots to individual, selfish goals. We encourage the reader to view our demos at <https://aalpatya.github.io/gbpstack>.

I. INTRODUCTION

Multi-robot teams offer the potential of solving complex tasks beyond the abilities of single robots, and for scalability it is desirable to seek distributed solutions where the robots operate with local computation and peer-to-peer communication rather than centralised control. The fact that robot teams are not yet widely used in real applications perhaps reflects the many aspects of competence that are needed to achieve whole tasks, and the difficulty in providing distributed, yet connected solutions to these. For instance a robot team tackling an environmental inspection task must be capable of localisation, planning, mapping and coordinated active information acquisition. Each one of these competences on its own is difficult to achieve in a distributed manner, but for the whole task to be performed well they must also influence each other: localisation is needed for planning, which in turn is needed for coordinated information acquisition.

In this work we show that various types of competence for robot teams can be modelled as *factor graphs*, and that this formulation allows for straightforward and principled coupling between competencies as layers in a graph stack.

The novelty of our method is to show that the various competencies or problem domains that influence each other can be optimised independently and asynchronously. Robots effectively hold and update a local copy of a global state based on ad-hoc collaboration with other robots, reaching consensus on its true value.

We build on recent work which showed that Gaussian Belief Propagation (GBP) is a powerful general tool for distributed factor graph inference, specifically when applied to multi-robot planning [1].

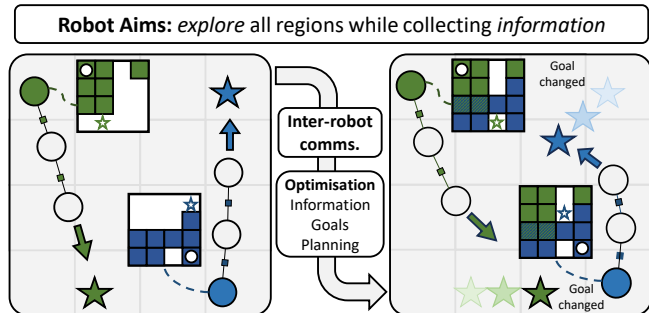


Fig. 1: A simplified schematic of the general applications we consider. Left: two robots (circles) have planned their paths towards goals (stars) in unexplored regions based on partial information they have estimated about the global state (shown here as a grid for each robot representing exploration). Right: after message passing they have combined and improved their estimates, optimising their planned paths towards new regions.

Although our formulation for such a joint-optimisation problem is general, we demonstrate it here in simulations of multi-robot exploration and information acquisition about a global state. In the first scenario a swarm of robots must search for a target region in the environment. In the second, the robots must collectively explore the whole environment and accurately estimate the global state. Finally we investigate the performance of our algorithm under communication limitations.

In summary, our main contributions are:

- The first application of GBP to a multi-robot, joint optimisation problem over multiple competencies where robots must reach a consensus on a global state through local measurements.
- Qualitative and quantitative simulations to show that our method outperforms related work in swarm source-seeking, and the scalability and robustness of our algorithm under communication limitations.

II. RELATED WORK

We are interested in planning for multi-robot teams which have coordinated goals, such as efficient, collaborative information acquisition. Information-based planning is a challenging problem with wide-ranging applications such as gas distribution mapping [2], environmental monitoring [3], or target search. Non-myopic information acquisition methods for single robots can be categorised into search-based and

sampling-based methods. Search-based methods can find globally optimal paths by searching the motion and information spaces at the expense of computational efficiency [4]. Sampling-based methods however are able to find solutions, but relax guarantees about optimality by considering subsets of the search space. The centralised algorithm in [5] explores the motion and information spaces by incrementally creating a directed tree, biasing sampling towards informative regions of the environment.

Distributed solutions for multiple robots offer the clear potential for much faster and more scalable information acquisition, but require coordination and communication. The semi-distributed algorithm for gas-plume tracking in [6] adjusts a swarm formation to maximise the combined perception of the swarm around areas of high gas concentrations. A centralised unit aggregates measurements made by robots and assigns tasks to each of them to maximise the combined perception of the plume around areas of high gas concentration. If global communication between robots can be assumed, algorithms based on Particle Swarm Optimisation (PSO) can be used which model robots as particles performing random walks biased towards the collective best observation regions. These methods do not guarantee success but have been shown to operate well in real-world experiments [7] where a swarm of small drones localise a gas source in an unknown cluttered environment by communicating observed gas concentrations with the group. Without the guarantee of global connectivity, efficient information dissemination techniques are needed once a robot has completed its task. The hybrid PSO method in [8] simulated robots exploring a complex environment in the form of Perlin noise, searching for a target region below a given threshold. Once a suitable region was found, robots aimed to maximise dissemination of the information across the group with minimal coordination.

These approaches do not consider robot dynamics or noisy signal measurements in their optimisation and produce unrealistic, jerky paths. When there are differences in the measurements by two robots of a global property due to sensor noise, they should be able to reach a consensus on the true value [9].

Multi-robot planning is a well studied topic in literature where robots must plan paths through an environment while avoiding each other and obstacles which may be dynamic. Local distributed planners optimise for the planned paths of robots in a short forward time window, and allow for efficient re-planning of paths in reaction to changes in the environment through communication with other robots [10], [11].

Recently, Gaussian Belief Propagation (GBP) has been shown to be an ideal tool for distribution of convergent computation between multiple robots, whether for multi-robot localisation [12] or short-horizon avoidance planning [1] and can offer efficient solutions. However, in that work, the multiple robots all had *separate*, selfish goals. In this new paper, for the first time we show that an integrated, multi-layer factor graph approach with GBP can be used to achieve coordinated multi-robot behaviour for information

acquisition. The different elements of the whole system — local avoidance planning, global information-guided goal coordination and coordinated, convergent map creation, can all be formulated as linked factor graphs.

III. TECHNICAL BACKGROUND

A. Factor Graphs

Factor graphs are a useful tool for representing sparse networks of interrelated variables and constraints. A factor graph is an undirected bipartite graph and represents the factorisation of a joint function $p(\mathbf{X})$ into components f_s which depend on subsets \mathbf{X}_s of all variables \mathbf{X} :

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

In a Gaussian factor graph all factors have the form of Gaussian distributions:

$$f_s(\mathbf{X}_s) \propto e^{-\frac{1}{2}[(z_s - \mathbf{h}_s(\mathbf{X}_s))^\top \Lambda_s (z_s - \mathbf{h}_s(\mathbf{X}_s))]} , \quad (2)$$

where $\mathbf{h}_s(\mathbf{X}_s)$ is the functional form of the constraint that the factor represents, z_s is its observed or expected value and Λ_s is the precision (inverse covariance) of the constraint.

Factor graph inference involves finding the values of variables \mathbf{X} which maximise $p(\mathbf{X})$. This is equivalent to finding \mathbf{X} to *minimise* the ‘energy’ $E(\mathbf{X}) = -\log p(\mathbf{X})$. For a Gaussian factor graph, this is effectively minimising a sum of squared terms:

$$E(\mathbf{X}) = L + \frac{1}{2} \sum_s (z_s - \mathbf{h}_s(\mathbf{X}_s))^\top \Lambda_s (z_s - \mathbf{h}_s(\mathbf{X}_s)) , \quad (3)$$

where L is a constant.

Multi-robot optimisation problems that can be formulated as least squares minimisation over the variables can be solved by performing inference on a Gaussian factor graph. Powerful factor graph inference tools which were originally developed for probabilistic inference could be used. They have been demonstrated for single robot planning using centralised solvers such as GTSAM [13], and for multi-robot planning in [1]. In this work we use a multi-layered factor graph to represent an abstraction for multi-robot planning problems that are dependent on information acquisition about a global state as illustrated in Figure 2. We will explain the details of the factors shown in the sections that follow.

B. Gaussian Belief Propagation (GBP)

GBP allows inference on Gaussian factor graphs via distributed computation. Storage of the graph can be shared between multiple devices communicating by radio or some other channel because convergence is achieved via node-wise computation and message passing around the graph. GBP is the special case of more general loopy belief propagation and has demonstrated excellent performance, obtaining exact solutions for the marginal means of all variables with rapid convergence [14]. GBP has been used [12] to solve the multi-robot problem of scalable localisation of large numbers of robots making noisy inter-robot measurements.

GBP proceeds by performing iterations of message passing between variables and factors around the loopy factor graph. The current marginal beliefs for variables can be obtained at any time. We now show the main steps; see [1], [12], [14] for full details.

We represent a Gaussian distribution in canonical/information form as:

$$\mathcal{N}(\mathbf{X}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}^{-1}(\mathbf{X}; \boldsymbol{\eta}, \boldsymbol{\Lambda}), \quad (4)$$

where $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$ are the precision matrix and information vector respectively. In GBP, variables \mathbf{X}_s are assumed to be Gaussian; thus, each variable \mathbf{x}_k has a belief $b(\mathbf{x}_k) = \mathcal{N}^{-1}(\mathbf{x}_k; \boldsymbol{\eta}_k, \boldsymbol{\Lambda}_k)$. Factors $F = \{f_s\}_{s=1:N_f}$ are Gaussian constraints between variables; the factor $f_s(\mathbf{X}_s)$ is an arbitrary (non-linear) function that connects variables \mathbf{X}_s .

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

$$b(\mathbf{x}_k) = \prod_{f \in n(\mathbf{x}_k)} \mathbf{m}_{f \rightarrow k}(\mathbf{x}_k), \quad (5)$$

where $n(\mathbf{x}_k) \subseteq F$ is the set of factors that the variable \mathbf{x}_k is connected to, and $\mathbf{m}_{f \rightarrow k}(\mathbf{x}_k) = \mathcal{N}^{-1}(\mathbf{x}_k; \boldsymbol{\eta}_{f \rightarrow k}, \boldsymbol{\Lambda}_{f \rightarrow k})$ is the message from a factor to the variable. In the canonical or information representation of Gaussians, this product can be rewritten as a summation:

$$\boldsymbol{\eta}_k = \sum_{f \in n(\mathbf{x}_k)} \boldsymbol{\eta}_{f \rightarrow k}, \quad \boldsymbol{\Lambda}_k = \sum_{f \in n(\mathbf{x}_k)} \boldsymbol{\Lambda}_{f \rightarrow k}. \quad (6)$$

2) *Variable to Factor Message*: A message from a variable \mathbf{x}_k to a factor $f_j \in n(\mathbf{x}_k)$ is the product of all incoming factor to variable messages apart from the message from f_j :

$$\mathbf{m}_{\mathbf{x}_k \rightarrow j}(f_j) = \prod_{f \in n(\mathbf{x}_k) \setminus f_j} \mathbf{m}_{f \rightarrow k}(\mathbf{x}_k). \quad (7)$$

3) *Factor Likelihood Update*: The likelihood of a factor $f_s(\mathbf{X}_s)$ connected to variables \mathbf{X}_s with linear measurement function $\mathbf{h}_s(\mathbf{X}_s)$, observation \mathbf{z}_s , and precision $\boldsymbol{\Lambda}_s$ can be expressed as a Gaussian distribution $\mathcal{N}^{-1}(\mathbf{X}_s; \boldsymbol{\eta}_f, \boldsymbol{\Lambda}_f)$, where $\boldsymbol{\eta}_f = \boldsymbol{\Lambda}_s(\mathbf{z}_s - \mathbf{h}_s(\mathbf{X}_s))$ and $\boldsymbol{\Lambda}_f = \boldsymbol{\Lambda}_s$. If the measurement function is non-linear, we use a first-order Taylor approximation to obtain the likelihood of the linearised factor which takes the form [15]:

$$\boldsymbol{\eta}_f = \mathbf{J}_s^\top \boldsymbol{\Lambda}_s (\mathbf{J}_s \mathbf{X}_s^0 + \mathbf{z}_s - \mathbf{h}_s(\mathbf{X}_s^0)), \quad (8)$$

$$\boldsymbol{\Lambda}_f = \mathbf{J}_s^\top \boldsymbol{\Lambda}_s \mathbf{J}_s, \quad (9)$$

where \mathbf{J}_s is the Jacobian and \mathbf{X}_s^0 is the linearisation point: the current state of the variables. In our work $\mathbf{z}_s = 0$ for all the factors unless stated otherwise, meaning that the factor energy is purely a function of the variables.

4) *Factor to Variable Message*: We take the product of the factor likelihood and messages from $\mathbf{X}_s \setminus \mathbf{x}_k$, then marginalise out all variables but \mathbf{x}_k resulting in a message from a factor to variable \mathbf{x}_k :

$$\mathbf{m}_{f \rightarrow k}(\mathbf{x}_k) = \sum_{\mathbf{x} \in \mathbf{X}_s \setminus \mathbf{x}_k} f_s(\mathbf{X}_s) \prod_{\mathbf{x} \in \mathbf{X}_s \setminus \mathbf{x}_k} \mathbf{m}_{\mathbf{x} \rightarrow f}(\mathbf{x}). \quad (10)$$

IV. METHOD

We consider the complex problem of exploration and information acquisition where a swarm of robots $\mathcal{R} = [\mathcal{R}_i, i \in \{0..N_R\}]$ must efficiently explore an environment $\mathcal{M} \in \mathbb{R}^d$ to collaboratively estimate a global state $\boldsymbol{\Psi}$.

In general $\boldsymbol{\Psi}$ can represent any scalar or vector quantity present throughout the environment with values in the range [0,1] of which robots can make noisy and partial measurements. In our work $d = 2$ and $\boldsymbol{\Psi}$ represents a scalar signal field such as gas concentration present throughout the environment \mathcal{M} which is segmented into N_M square sampling regions of width r_D m. Robots can make measurements of $\boldsymbol{\Psi}$ for regions $m \in \mathcal{M}(r_S)$ within their sampling radius r_S .

The aims of the robots are to:

- Collaborate with other robots to form a consensus on the global state using noisy local measurements.
- Find a location where the signal value is below a given threshold, or collectively explore the whole environment.
- Avoid collisions with other robots and plan smooth paths over a time horizon T_H in Euclidean space.

A simplified schematic of the problem is shown in Figure 1.

A. A GBP Stack of Factor Graphs

Robots plan paths towards intermediate goal locations in the environment which directly influence the information they collect about $\boldsymbol{\Psi}$ through sampling. However, this collected information can also actively guide the path planning goals to new and potentially informative regions; there is a clear inter-dependency between information acquisition, path planning and goal selection.

We formulate the problem as a *stack of factor graphs* which we call the ‘GBP stack’, where the layers of the stack represent different problem domains that can be solved using GBP. Each robot holds variables and factors in the layers of its own GBP stack, and is able to create and destroy factors between themselves and their neighbours in an ad-hoc manner. In this way optimisation can be done using GBP both *within* a robot’s stack as well as *between* a robot and its neighbours.

This is a general formulation, but in this work we consider a robot’s GBP stack formed of layers representing information acquisition, goal selection, and local path planning as shown in Figure 2. The information layer \mathcal{I} contains a robot’s copy of the global state $\boldsymbol{\Psi}$, updated with local measurements and information from other robots. The goal layer \mathcal{G} is responsible for the selection of regions to plan paths towards. Finally the planning layer \mathcal{P} is responsible for the short range path planning and collision avoidance of the robot.

In real-world systems robots may be restricted in their inter-robot communication due to bandwidth or power constraints and may not receive information at every timestep for each layer. The asynchronous nature of GBP allows a robot to continue optimising over \mathcal{G} and \mathcal{P} using the most recent information it holds in \mathcal{I} .

We now describe in each layer in detail, and the types of variables and factors that are involved.

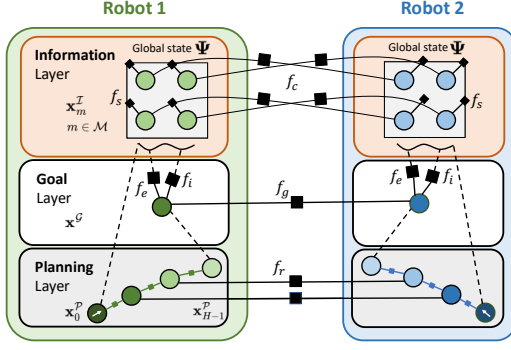


Fig. 2: The GBP stack of factor graphs for two robots estimating a global state Ψ shown here as being segmented into $N_M = 4$ regions. A robot holds a copy of Ψ in its Information layer which it can update asynchronously. The Goal layer guides the Planning layer for exploration. Message passing can happen asynchronously between the layers of a robot's stack as well the stacks of other robots.

B. Information layer \mathcal{I}

This layer represents a robot's copy of the global state; there is one variable $\mathbf{x}_m^{\mathcal{I}} = [\mathbf{p}_m^{\mathcal{I}}, \psi_m^{\mathcal{I}}, \zeta_m^{\mathcal{I}}]^{\top}$ for each region m in the environment. Dropping the superscript \mathcal{I} for brevity, this variable is formed of the position \mathbf{p}_m , signal value ψ_m , and coverage $\zeta_m \in [0, 1]$ of the region m . $\zeta_m = 1$ signifies that the cell has been visited by the robot, and $\psi_m = 1$ at initialisation.

1) *Sensor Factor*: Each variable is connected to a unary factor representing the noisy sensor measurement \mathbf{z}_s made by the robot of the region m . The factor is represented as

$$\mathbf{f}_s : \mathbf{h}_s(\mathbf{x}_m^{\mathcal{I}}) = \mathbf{x}_m^{\mathcal{I}}, \quad \Lambda_s = \text{diag}(\sigma_p^{-2}, \sigma_\psi^{-2}, \sigma_\zeta^{-2}), \quad (11)$$

$$\mathbf{z}_s \sim \mathcal{N}([\tilde{\mathbf{p}}_m, \tilde{\psi}_m, 1]^{\top}, \Lambda_s^{-1}). \quad (12)$$

where the tilde \sim above a variable represents its true value. In our work σ_p and $\sigma_\zeta = 10^{-5}$, representing perfect measurement of the location of m .

2) *Inter-robot Consensus Factor*: Each variable in this layer is also connected to the corresponding variables in the Information layers of other robots. These factors have precision $\Lambda_c = \sigma_c^{-2}\mathbf{I}$ and are represented as

$$\mathbf{f}_c : \mathbf{h}_c(\mathbf{x}_m^{\mathcal{I}_1}, \mathbf{x}_m^{\mathcal{I}_2}) = \mathbf{x}_m^{\mathcal{I}_1} - \mathbf{x}_m^{\mathcal{I}_2}. \quad (13)$$

For a pair of robots (R_1, R_2), these factors encourage R_1 's beliefs of the variables to be the same as those of R_2 , allowing the robots to reach a consensus on the global state Ψ through iterations of GBP, as well as exchanging information about the regions of the environment they have collectively visited.

C. Goal layer \mathcal{G}

The Goal layer in a robot's GBP stack is responsible for deciding on regions to plan paths towards whilst avoiding regions that other robots are heading to. Each robot holds a variable in this layer $\mathbf{x}^{\mathcal{G}}$ representing a goal location for the robot's path planning layer. This variable is connected to factors representing information acquisition, exploration and inter-robot coordination.

1) *Signal Factor*: A factor \mathbf{f}_i aims to draw the goal variable to the region m^* with the best (lowest) value ψ_{m^*} of Ψ . It is therefore dependent on the most up-to-date knowledge from the Information layer. The factor precision matrix is $\Lambda_i = \sigma_i^{-2}\mathbf{I}$ and the factor is denoted as

$$\mathbf{f}_i : \mathbf{h}_i(\mathbf{x}^{\mathcal{G}}; \mathbf{p}_{m^*}^{\mathcal{I}}) = u(m^*)(\mathbf{x}^{\mathcal{G}} - \mathbf{p}_{m^*}^{\mathcal{I}}), \quad (14)$$

$$u(m^*) = 1 - \psi_{m^*}, \quad (15)$$

$$m^* = \underset{m \in \mathcal{M}}{\text{argmin}} \psi_m. \quad (16)$$

2) *Exploration Factor*: This factor \mathbf{f}_e aims to draw the goal variable $\mathbf{x}^{\mathcal{G}}$ towards the nearest unexplored region and is therefore also dependent the Information layer as well the Planning layer with the robot's current position $\mathbf{p}_0^{\mathcal{P}}$. The factor precision is $\Lambda_e = \sigma_e^{-2}\mathbf{I}$ and has the same form as (14) with

$$u(m^*) = 1, \quad m^* = \underset{m \in \{\underset{m' \in \mathcal{M}}{\text{argmin}} \zeta_{m'}\}}{\text{argmin}} \|\mathbf{p}_{m^*}^{\mathcal{I}} - \mathbf{p}_0^{\mathcal{P}}\|. \quad (17)$$

If there are no unexplored regions, a random location is selected so that robots continue to explore.

3) *Goal Diversity Factor*: Finally $\mathbf{x}^{\mathcal{G}}$ is connected to the goal variables of neighbouring robots. This discourages two connected robots from moving towards the same region and therefore promotes exploratory behaviour in the swarm. The factor has precision $\Lambda_g = \sigma_g^{-2}\mathbf{I}$ and is formed as

$$\mathbf{f}_g : \mathbf{h}_g(\mathbf{x}^{\mathcal{G}_1}, \mathbf{x}^{\mathcal{G}_2}) = \begin{cases} 1 - \frac{\|\mathbf{x}^{\mathcal{G}_1} - \mathbf{x}^{\mathcal{G}_2}\|}{r_D} & \|\mathbf{x}^{\mathcal{G}_1} - \mathbf{x}^{\mathcal{G}_2}\| \leq r_D \\ 0 & \text{otherwise} \end{cases}. \quad (18)$$

D. Planning layer \mathcal{P}

This layer is based on our previous work [1] where a robot's path consisted of variables representing its position and velocity in a short forward time window T_H . These variables were connected with factors representing dynamics \mathbf{f}_d and collision avoidance \mathbf{f}_r , and the planning horizon state had a fixed non-optimisable velocity.

In this work however the horizon state $\mathbf{x}_{H-1}^{\mathcal{P}}$ is dependent on the Goal layer, having a velocity of magnitude V_{max} towards $\mathbf{x}^{\mathcal{G}}$. A robot's planned path can therefore dynamically change based on the latest information of the world.

E. Algorithm

Robots initialise their GBP stacks with the variables and factors in each layer as described in sections IV-B to IV-D. At every timestep, each robot follows Algorithm 1.

When a new robot R_j is within communication range r_C of robot R_i , inter-robot factors are created between the two

Algorithm 1 For each robot R_i

- 1: Initialise GBP stack layers $\mathcal{I}, \mathcal{G}, \mathcal{P}$
 - 2: **while** $t < T_{max}$ **do**
 - 3: Let $N(R_i) = \{R_j \mid \|\mathbf{p}_0^{P_i} - \mathbf{p}_0^{P_j}\| < r_C\}$ be the set of robots within the communication radius of R_i .
 - 4: Let $C(R_i)$ be the set of robots connected to R_i .
 - 5: **for** Newly observed robot $R_j \in N(R_i) \setminus C(R_i)$ **do**
 - 6: Create inter-robot factors f_c, f_g, f_r .
 - 7: **for** Out-of-range robot $R_j \in C(R_i) \setminus N(R_i)$ **do**
 - 8: Delete inter-robot factors f_c, f_g, f_r .
 - 9: Make local measurement of $\mathbf{x}_m^{\mathcal{I}} \forall m \in \mathcal{M}(r_S)$
 - 10: Activate variables $\mathbf{x}_m^{\mathcal{I}} \forall m \in \mathcal{M}(r_C)$.
 - 11: **for** layer \mathcal{L} in GBP Stack **do**
 - 12: Perform N_I iterations of GBP
 - 13: Update $\mathbf{x}_0^{\mathcal{P}}$ and $\mathbf{x}_{H-1}^{\mathcal{P}}$ by ΔT .
-

robots for each layer in the robots' GBP stacks if they do not already exist, and are destroyed when the robots are out of range.

The robot samples the environment for regions $m \in \mathcal{M}(r_S)$ and updates its corresponding Information layer variables $\mathbf{x}_m^{\mathcal{I}}$. Joint optimisation is performed over all layers using GBP for N_I iterations. In the Information layer, only variables representing regions within the robot's communication radius r_C are set to 'active' and take part in inter-robot message passing. Finally the robot's position and horizon states in its planning layer are updated using the simulation timestep ΔT .

V. EXPERIMENTS

We model a swarm of N_R robots with radius $r_R = 1\text{m}$ and planning horizon $T_H = 1\text{s}$ traversing a square environment of length D m, segmented into cells with side length $r_D = 10\text{m}$. In the experiments that follow, the robots have a maximum speed of $V_{max} = 5\text{m/s}$, and make measurements every 1s within a sampling radius $r_S = 10\text{m}$. The communication intervals for the Goal and Information layers of the robots' GBP stacks are set to $T_C^{\{\mathcal{I}, \mathcal{G}\}} = 1\text{s}$, and $T_C^{\mathcal{P}} = \Delta T = 0.1\text{s}$. We set $N_I = 5$, $\sigma_c = 1$ [-], $\sigma_e = 0.1$ m, $\sigma_i = 0.5\text{m}$ and $\sigma_g = 0.01$ [-], encouraging exploration whilst seeking regions of low ψ .

A. Baseline for Comparison

We compare our method with the Hybrid Particle Swarm Optimisation (PSO) algorithm of [8], where a swarm of robots explores a complex environment with a scalar signal field searching for a region with signal value less than a given threshold. Similar to our work, the authors consider a dynamic graph topology of robot connectivity and the method operates in a purely decentralised manner. However it does not account for robot collision avoidance and creates unrealistic jerky paths. For a fair comparison we use this work as a baseline in the source seeking experiment to show the validity of our method against the same complexities

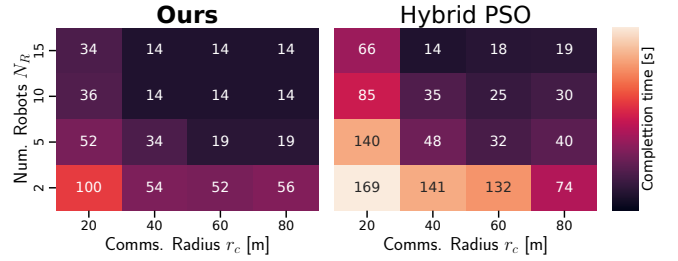


Fig. 3: Completion times for the source seeking experiment as the number of robots N_R and communication radius r_C varies. Our method is faster than Hybrid PSO while being able to consider noisy sensor measurements.

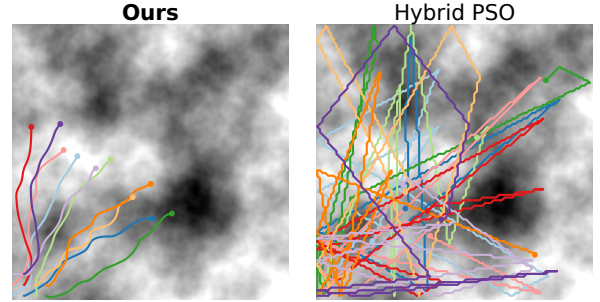


Fig. 4: Paths taken until the target region was found in the source seeking experiment for $N_R = 10$ robots and communication radius $r_C = 40$ m. Our method optimises for smooth paths and efficient coverage of the environment without collisions.

of 2D Perlin noise environments and initial robot configurations. We use the best performing parameters from the work: $c_{p,g} = 1, \omega = 0$.

B. Source Seeking

Robots begin in one corner of the environment with $D = 100\text{m}$ and search for a region where the value of the global state ψ is less than a threshold $\psi^* = \frac{10}{255}$ (i.e. a 'source'). The noise on their measurements of ψ is set to $\sigma_\psi = 0.01$ for our method while Hybrid PSO assumes perfect sensing.

We measure the time taken for all robots to have found or gained knowledge of the target region as we vary the number of robots N_R and the radius of communication r_C .

Figure 3 shows that as N_R and r_C increase the completion time decreases as robots quickly spread and cover a larger area of the environment, due to their goal diversity factors f_g as well as being able to share information about distant regions of the environment. A larger N_R results in diminishing returns for completion times due to the increase in robot density in space. Our method outperforms the baseline due to its efficient information sharing and path planning and is able to create smooth paths as seen in Figure 4.

C. Exploration and Coverage

In this experiment, the task of the swarm is to collectively explore an environment of width $D = 200$ m, taking noisy local measurements of the global signal field Ψ with

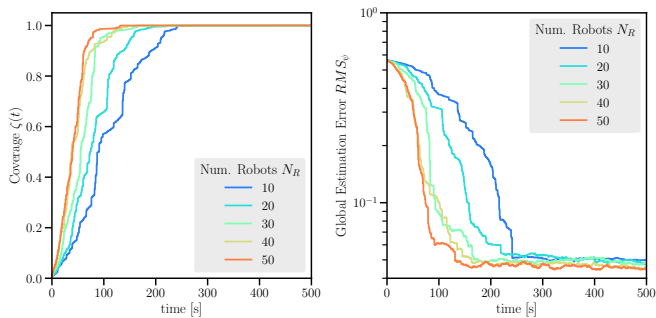


Fig. 5: Evolution of coverage and RMS_{ψ} over time for the ‘random’ initialisation scenario. Increasing the number of robots N_R results in quicker coverage and a steady-state RMS_{ψ} much lower than the measurement noise ($\sigma_s = 0.1$).

$\sigma_{\psi} = 0.1$. We weaken the signal factor $\sigma_i = 1000$ whilst keeping the other factors of the same strength to encourage exploration. We vary the number of robots N_R in the swarm and consider the evolution of two metrics over time:

Coverage $\zeta(t)$: the proportion of the environment visited ($\zeta_m > 0$) by a robot, averaged over all robots.

Global Estimation Error $RMS_{\psi}(t)$: the root-mean-square difference between a robot’s beliefs of ψ_m and the true values $\tilde{\psi}_m$ averaged over all regions $m \in \mathcal{M}$ and all robots.

Robots are initialised in random positions with zero velocity, and Figure 5 shows that increasing the number of robots N_R results in faster coverage; robots perform goal layer optimisation and plan paths towards distinct regions. The steady-state values of RMS_{ψ} decrease as N_R increases, and are significantly lower than the standard deviation of noise $\sigma_{\psi} = 0.1$ on robots’ sensors.

D. Varying Communications Radius

When two robots communicate they share information about regions of the environment that are within r_C of their current position. This is useful for scalability as it limits the amount of data transferred between them (robots do not need to share information about their whole map). For the two initial conditions ‘corner’ and ‘random’ we measure the time taken for full coverage as well as the steady-state value of RMS_{ψ} after 1000s for representative low, medium and large r_C .

Table I shows that a large r_C results in faster coverage and lower RMS_{ψ} as robots are able to process information about distant regions and can benefit from being spread out as in the ‘random’ scenario. For small r_C , the ‘corner’ scenario (see Figure 4) performs better – robots begin close together resulting in more inter-robot communication as compared to the ‘random’ scenario.

Although a larger communication radius is beneficial for the swarm, this effectively increases the amount of data transferred and presents a trade-off for real-world systems.

E. Performance under Communication Limitations

A key aspect of our multi-layer approach is that robots do not constantly need to take part in information exchange.

TABLE I: Effect of varying communication range r_C for $N_R = 20$ in the ‘corner’ and ‘random’ initialisations

r_C	Coverage Time [s]		Steady-state RMS_{ψ}	
	Corner	Random	Corner	Random
20	147	232	0.0408	0.0425
50	84	56	0.0375	0.0381
100	78	48	0.0359	0.0343

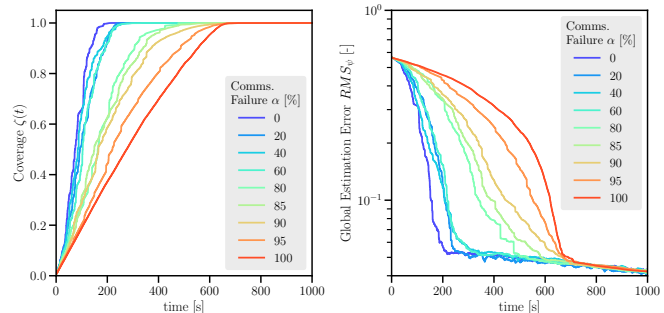


Fig. 6: Evolution of coverage and RMS_{ψ} over time as the communications failure rate α varies. Robots take longer to explore the environment as inter-robot communication decreases whilst still accurately estimating the global state.

Real-world robots may be subject to power limitations or sensor failures, and may prioritise collision avoidance over information acquisition. We simulate this regime of communications failure for $N_R = 20$ robots in the ‘random’ initialisation scenario. At every timestep a proportion α of randomly selected robots *do not* take local measurements or perform inter-robot communication in their Goal and Information layers.

Figure 6 shows that as α increases it takes longer for the swarm to cover the full environment and to accurately estimate Ψ as robots cannot efficiently decide on goal locations for path planning, or make use of their neighbours’ collected information. Robots continue to optimise over their own GBP stack, making use of new information when it becomes available; full coverage and accurate estimation is still achieved when the robots behave independently ($\alpha = 1$).

VI. CONCLUSIONS

We have shown that our distributed method for collaborative optimisation over multiple problem competencies can enable a team of robots to accurately estimate a global state based purely on local independent measurements. By formulating the problem as a stack of linked factor graphs robots optimise over each problem competence in an asynchronous manner. The distributed nature of our algorithm promises scalability to large environments and numbers of robots. In future work we will test our algorithm with more complex maps, and investigate optimisation with a dynamic model of the global state as here we have assumed it to be constant with time.

ACKNOWLEDGEMENTS

This research has been supported by the EPSRC Prosperity Partnership Award with Dyson Technology Ltd.

REFERENCES

- [1] 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.
- [2] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J. Pappas. Information acquisition with sensing robots: Algorithms and error bounds. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6447–6454, 2014.
- [3] Bradley Woosley, Prithviraj Dasgupta, John G. Rogers, and Jeffrey Twigg. Multi-robot information driven path planning under communication constraints. *Autonomous Robots*, 44(5):721–737, May 2020. Publisher Copyright: © 2019, Springer Science+Business Media, LLC, part of Springer Nature.
- [4] Jerome Le Ny and George J. Pappas. On trajectory optimization for active sensing in gaussian process models. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 6286–6292, 2009.
- [5] Yiannis Kantaros, Brent Schlotfeldt, Nikolay Atanasov, and George Pappas. Sampling-based planning for non-myopic multi-robot information gathering. *Autonomous Robots*, 45:1–18, 10 2021.
- [6] A. Ch. Kapoutsis, I. T. Michailidis, Y. Boutalis, and E. B. Kosmatopoulos. Building synergetic consensus for dynamic gas-plume tracking applications using uav platforms. *Computers & Electrical Engineering*, 91, 2021.
- [7] B. P. Duisterhof, S. Li, J. Burgues, V.J. Reddi, and G.C.H.E. de Croon. Sniffy bug: A fully autonomous swarm of gas-seeking nano quadcopters in cluttered environments. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [8] Julia T. Ebert, Florian Berlinger, Bahar Haghighat, and Radhika Nagpal. A hybrid pso algorithm for multi-robot target search and decision awareness. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11520–11527, 2022.
- [9] Aniket Shirsat, Karthik Elamvazhuthi, and Spring Berman. Multi-robot target search using probabilistic consensus on discrete markov chains. In *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 108–115, 2020.
- [10] Xin Zhou, Jiangchao Zhu, Hongyu Zhou, Chao Xu, and Fei Gao. Ego-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4101–4107, 2021.
- [11] C. Luis, M. Vukosavljev, and A. Schoellig. Online trajectory generation with distributed model predictive control for multi-robot motion planning. *IEEE Robotics and Automation Letters*, PP:1–1, 01 2020.
- [12] R. Murai, J. Ortiz, S. Saeedi, P.H.J. Kelly, and A. J. Davison. A robot web for distributed many-device localisation. *arXiv preprint arXiv:2202.03314*, 2022.
- [13] F. Dellaert. Factor graphs: Exploiting structure in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:141–166, 2021.
- [14] J. Ortiz, T. Evans, and A. J. Davison. A visual introduction to Gaussian Belief Propagation. *arXiv preprint arXiv:2107.02308*, 2021.
- [15] A. J. Davison and J. Ortiz. FutureMapping 2: Gaussian Belief Propagation for Spatial AI. *arXiv preprint arXiv:1910.14139*, 2019.