# Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths

Tal Regev and Vadim Indelman

***Abstract*— In this paper we develop a new approach for decentralized multi-robot belief space planning in high-dimensional state spaces while operating in unknown environments. State of the art approaches often address related problems within a sampling based motion planning paradigm, where robots generate candidate paths and are to choose the best paths according to a given objective function. As exhaustive evaluation of all candidate path combinations from different robots is computationally intractable, a commonly used (sub-optimal) framework is for each robot, at each time epoch, to evaluate its own candidate paths while only considering the best paths announced by other robots. Yet, even this approach can become computationally expensive, especially for high-dimensional state spaces and for numerous candidate paths that need to be evaluated. In particular, upon an update in the announced path from one of the robots, state of the art approaches re-evaluate belief evolution for *all* candidate paths and do so from scratch. In this work we develop a framework to identify and efficiently update *only* those paths that are actually impacted as a result of an update in the announced path. Our approach is based on appropriately propagating belief evolution along impacted paths while employing insights from factor graph and incremental smoothing for efficient inference that is required for evaluating the utility of each impacted path. We demonstrate our approach in a synthetic simulation.**

## I. INTRODUCTION

Collaboration between multiple robots pursuing common or individual tasks is important in numerous problem domains, including cooperative navigation, mapping, tracking and active sensing. A key required capability is to autonomously determine robot actions while taking into account different sources of uncertainty.

The corresponding problem can be formulated within a partially observable Markov decision process (POMDP) framework, which is known to be computationally intractable [20]. Thus, the research community has been extensively investigating approximate approaches to provide better scalability to support real world problems. These approaches can be roughly classified into four categories, some of which are further discussed below: point-based value iteration methods (e.g. [16]), simulation based approaches (e.g. [24]) in the context of active SLAM, sampling based approaches (e.g. [14], [15], [17]) and direct trajectory optimization approaches (e.g. [12], [22], [25]).

T. Regev is with the Department of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel, talregev@technion.ac.il. V. Indelman is with the Department of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel, vadim.indelman@technion.ac.il. This work was partially supported by the Technion Autonomous Systems Program (TASP) and by the Ministry of Science & Technology, Israel & the Russian Foundation for Basic Research, the Russian Federation.

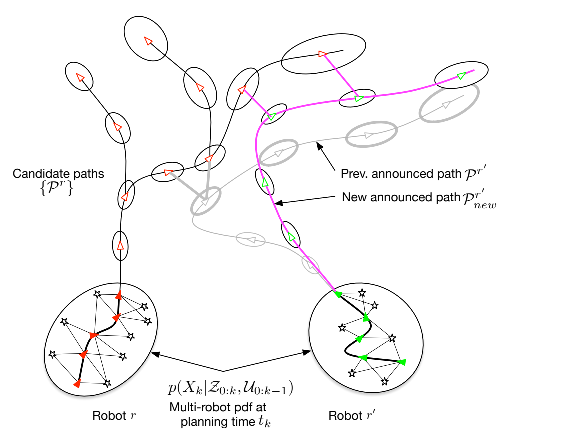



**Fig. 1:** Illustration of the proposed concept. The figure shows belief evolution over a few candidate paths of robot $r$ given an announced path $\mathcal{P}^{r'}$ from robot $r'$ and the corresponding multi-robot constraints that can represent, e.g., future mutual observations of environments unknown at planning time [10]. Upon an update in an announced path from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$, a new set of such constraints will be generated (shown in purple), requiring to re-calculate belief evolution for candidate paths. Covariance ellipses are shown for illustration.

In particular, sampling based approaches (e.g. [1], [5], [9], [23]) discretize the state space using randomized exploration strategies to explore the belief space in search of an optimal plan. While many of these approaches, including probabilistic roadmap (PRM) [15], rapidly exploring random trees (RRT) [17], and RRT* and Rapidly-exploring Random Graph (RRG) [14], assume perfect knowledge of the state, deterministic control and a known environment, efforts have been devoted in recent years to alleviate these restricting assumptions. The corresponding approaches include, for example, the belief roadmap (BRM) [23] and the rapidly-exploring random belief trees (RRBT) [5], where planning is performed in the belief space, thereby incorporating the predicted uncertainties of future position estimates. Similar strategies are used to address also informative planning problems (see, e.g., [9]).

While typically the environment is assumed to be known, recent research focused on facilitating autonomous operation also in the presence of uncertainty in the environment and when the environment is a priori unknown and instead is mapped on the fly, see e.g. [6], [12], [24]. The problem is tightly related to active SLAM and can be formulated within POMDP framework.

A multi-robot belief space framework has been also investigated in different contexts in recent years, including multi-robot tracking, active SLAM and autonomous navigation in unknown environments, planning for coverage tasks, and informative planning (see, e.g. [3], [10], [11], [18]). In particular, in a recent work [10] we considered the problem of multi-robot active collaborative estimation while operating in unknown environments and introduced within the belief reasoning regarding future mutual observations of environments that are unknown at planning time. Here, we build upon that work considering a decentralized framework.

Unfortunately, solving exactly the corresponding decentralized POMDP problem is computationally intractable and has been shown to be nondeterministic exponential (NEXP) complete [4], and thus has been typically addressed using approximate approaches. Also, despite the intractable worse case complexity of decentralized POMDP, there has been impressive progress in recent years in solving interesting instances of the problem (e.g. [2]).

A common approach to reduce computational complexity is for each robot, at each time epoch, to solve the belief space planning problem considering its own candidate paths (generated, e.g., by some sampling method) and the best solutions found and announced by other robots (e.g. [3], [18]). The robot then announces its best path, according to a user-defined objective function, to other robots which then proceed with the same procedure. Such an approach avoids solving the problem jointly over all robots and reduces the exponential complexity in the number of robots to linear complexity, with performance guarantees analyzed in [3].

Yet, existing methods calculate the belief evolution over *all* candidate paths from *scratch* each time a new announced plan from another robot is received, which by itself can be computationally extensive operation.

*Contribution:* In this work we contribute a multi-robot belief space planning approach which further reduces computational complexity, considering the problem of multi-robot autonomous navigation in unknown environments. Instead of re-evaluating from scratch each candidate path, the key observation is that often, belief evolution changes only for part of the candidate paths as a result of an update in the announced path from another robot(s). We show how to identify and efficiently recalculate *only* those candidate paths that are impacted as a result of an update in the announced paths from another robot. See illustration in Figure 1. Our approach is based on appropriately propagating belief evolution along impacted paths while employing insights from factor graph for efficient inference that is required for evaluating the utility of each impacted path.

## II. Probabilistic Formulation and Notations

We consider a group of $R$ robots operating in unknown or uncertain environments, aiming to autonomously decide their future actions based on information accumulated thus far and a given objective function $J$, which is a function of robots' beliefs at different future time instances.

Let $\mathbb{P}(X_k^r|\mathcal{Z}_{0:k}^r, \mathcal{U}_{0:k-1}^r)$ represent the posterior probability distribution function (pdf) at planning time $t_k$ over states of interest $X_k^r$ of robot $r$ (e.g. current and past poses). Here, $\mathcal{Z}_{0:k}^r$ and $\mathcal{U}_{0:k-1}^r$ denote, respectively, all observations and

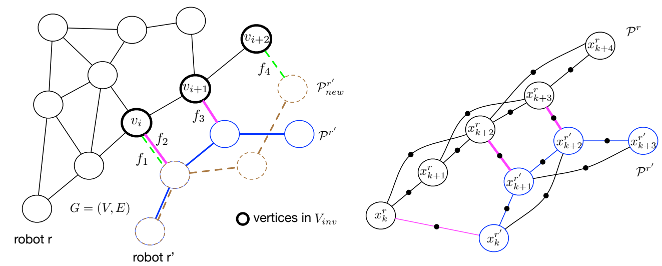



**Fig. 2:** (*left*) Graph $G = (V, E)$ along which different candidate paths $\mathcal{P}^r$ of robot $r$ can be defined. Announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ from robot $r'$ facilitate multi-robot factors $f_1$, $f_2$, $f_3$ and $f_4$. (*right*) An example of a factor graph representing the joint belief $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ for some candidate path $\mathcal{P}^r$. Different factor graphs are obtained for each path $\mathcal{P}^r$ considering either $\mathcal{P}^{r'}$ or $\mathcal{P}^{r'}_{new}$.

controls by time $t_k$. Consider conventional state transition and observation models

$$x_{i+1} = f(x_i, u_i, w_i)\ ,\ \ z_{i,j} = h(x_i, x_j, v_{i,j}) \tag{1}$$

with zero-mean Gaussian process and measurement noise $w_i \sim N(0, \Omega_w)$ and $v_{i,j} \sim N(0, \Omega_{vij})$, and with known information matrices $\Omega_w$ and $\Omega_{vij}$. Denoting the corresponding probabilistic terms to Eq. (1) by $\mathbb{P}(x_i|x_{i-1}, u_{i-1})$ and $\mathbb{P}(z_{i,j}|x_i, x_j)$, the pdf $\mathbb{P}(X_k^r|\mathcal{Z}_{0:k}^r, \mathcal{U}_{0:k-1}^r)$ can be written as

$$\mathbb{P}(X_k^r|\mathcal{H}_k^r) \propto \mathbb{P}(x_0^r) \prod_{i=1}^{k} \mathbb{P}(x_i^r|x_{i-1}^r, u_{i-1}^r) p(Z_i^r|X_i^r) \tag{2}$$

where the history $\mathcal{H}_k^r$ is defined as $\mathcal{H}_k^r \doteq \{\mathcal{Z}_{0:k}^r, \mathcal{U}_{0:k-1}^r\}$.

The measurement likelihood term $\mathbb{P}(Z_i^r|X_i^r)$ can be expanded in terms of individual observations, $\mathbb{P}(Z_i^r|X_i^r) = \prod_{j=1}^{n_i} \mathbb{P}(z_{i,j}^r|X_{i,j}^r)$. Here, $Z_i^r \doteq \{z_{i,j}^r\}_{j=1}^{n_i}$ and $n_i$ denotes the number of observations acquired at time $t_i$ and $X_{i,j}^r \subseteq X_i^r$ represents involved variables in the $j$th observation model. Note this formulation assumes known data association and does not consider outliers. Robust perception approaches do exist, however, both in inference (e.g. [19]) and, recently, in belief space planning [21].

We now consider all the $R$ robots in the group, and let $\mathbb{P}(X_k|\mathcal{H}_k)$ represent the pdf over the joint state $X_k$ at time $t_k$, where $X_k \doteq \{X_k^r\}_{r=1}^R$ and $\mathcal{H}_k \doteq \{\mathcal{Z}_{0:k}, \mathcal{U}_{0:k-1}\}$, with $\mathcal{Z}_{0:k} \doteq \{\mathcal{Z}_{0:k}^r\}_{r=1}^R$ and $\mathcal{U}_{0:k-1} \doteq \{\mathcal{U}_{0:k}^r\}_{r=1}^R$.

Let $J$ denote a user-defined objective function $J(\mathcal{U}) = \mathbb{E}\left[\sum_{l=1}^{L} c_l(b[X_{k+l}], u_{k+l})\right]$, where $u_{k+l} \doteq \{u_{k+l}^r\}$ and the expectation is taken with respect to future observations of all robots, and where $c_l$ represents an immediate cost function at the $l$th look ahead step, which can be a function of the joint belief $b[X_{k+l}]$ (to be defined) and of the controls. For simplicity, we use the same planning horizon $L$ for all robots.

In this paper we consider a special case of the objective function $J$ and assume the latter is of the following form:

$$J(\mathcal{U}) = \mathbb{E}\left[\sum_{l=1}^{L}\sum_{r=1}^{R} c_l^r(b[X_{k+l}^r], u_{k+l}^r)\right], \tag{3}$$

where $b[X_{k+l}^r] = \int_{\neg X_{k+l}^r} b[X_{k+l}]$ and thus depends on the multi-robot belief $b[X_{k+l}]$. Such a form naturally supports *collaborative active* state estimation, where each robot aims to improve its estimation accuracy while considering additional terms in $c_l$, if exist (see e.g. [11]).

In this paper, our objective is to find the optimal controls $\mathcal{U}^\star = \arg\min_{\mathcal{U}} J(\mathcal{U})$ for all robots in the group, considering a multi-robot decentralized framework discussed below.

## III. Decentralized Sampling-Based Planning

We consider a decentralized framework, where each robot calculates candidate paths using one of the existing sampling-based motion planning approaches (e.g. RRT, RRG, PRM). Adopting typical notations in literature, let $G^r = (V^r, E^r)$ be a graph maintained by robot $r$, with vertices $V^r$ representing sampled robot states and edges $E^r$ denoting feasible paths between corresponding vertices. Each vertex $v \in V^r$ is associated with a set of belief nodes, with each belief node representing a path $\mathcal{P}^r \doteq \{v_0, \ldots, v\}$ from the initial vertex $v_0$ that could be followed to reach the vertex $v$.

In this paper we interchangeably use $\mathcal{P}^r$ to represent a path and, when clear from context, also the corresponding robot states along that path. Denoting the state at each vertex $v$ by $x_v$, the corresponding joint belief over the entire path $\mathcal{P}^r$, considering for now only a single robot $r$, is

$$b[\mathcal{P}^r] \doteq \mathbb{P}(X_k^r, x_{v_0}^r, \ldots, x_v^r | \mathcal{H}_k^r, U(\mathcal{P}^r), Z(\mathcal{P}^r)), \quad (4)$$

where $U^r(\mathcal{P}^r)$ and $Z^r(\mathcal{P}^r)$ represent, respectively, the corresponding controls and (unknown) observations to be acquired by following the path $\mathcal{P}^r$. This pdf can be explicitly written in terms of the belief at planning time and the corresponding state transition and observation models as (see Eq. (2))

$$b[\mathcal{P}^r] = \mathbb{P}(X_k^r | \mathcal{H}_k^r) \mathbb{P}(\mathcal{P}^r | U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)), \quad (5)$$

where, for convenience, the local information (factors) along path $\mathcal{P}^r$ is defined as

$$FG_{local}(\mathcal{P}^r) \doteq \prod_{l=1}^{L(\mathcal{P}^r)} \mathbb{P}(x_{v_l}^r | x_{v_{l-1}}^r, u_{v_{l-1}}^r) \mathbb{P}(Z_{v_l}^r | X_{k+l}^r). \quad (6)$$

Throughout the paper we will often use the factor graph graphical model to represent a pdf. The factor graph for the pdf from Eq. (6) is denoted by $FG_{local}(\mathcal{P}^r)$.

The measurement likelihood term $\mathbb{P}(Z_{v_l}^r | X_{k+l}^r)$ can be further expanded, similarly to Eq. (2). Here, $X_{k+l}^r$ is the joint state up to the $l$th vertex along the path $\mathcal{P}^r$, i.e.:

$$X_{k+l}^r = X_{k+l}^r(\mathcal{P}^r) \equiv \mathcal{P}_{k+l}^r \doteq \{X_k^r, x_{v_0}^r, \ldots, x_{v_l}^r\}. \quad (7)$$

We now proceed to the multi-robot case and consider different paths $\mathcal{P}^r$ for each robot $r \in \{1, \ldots, R\}$. Letting $\mathcal{P} \doteq \{\mathcal{P}^r\}_{r=1}^R$, the multi-robot belief is given by

$$b[\mathcal{P}] = \mathbb{P}(X_k | \mathcal{H}_k) \prod_{r=1}^{R} \left[ \prod_{l=1}^{L(\mathcal{P}^r)} \mathbb{P}(x_{v_l}^r | x_{v_{l-1}}^r, u_{v_{l-1}}^r) \cdot \mathbb{P}(Z_{v_l}^r | X_{k+l}^r) \prod_{\{i,j\}} \mathbb{P}(z_{i,j}^{r,r'} | x_{v_i}^r, x_{v_j}^{r'}) \right], \quad (8)$$

where the last product corresponds to multi-robot constraints that can involve different time instances, representing mutual observations of a scene. With a slight abuse of notation, we use $x_{v_i}^r$ and $x_{v_j}^{r'}$ in the measurement likelihood term $\mathbb{P}(z_{i,j}^{r,r'} | x_{v_i}^r, x_{v_j}^{r'})$ to represent both a robot state before planning time, i.e. $x_{v_i}^r \subset X_k^r \subseteq X^r(\mathcal{P}^r)$ (likewise for $x_{v_j}^{r'}$), and a future state along the path $\mathcal{P}^r$. The latter case corresponds to a mutual observation of an area that is unknown at planning time, as introduced in our previous work [10].

The index set $\{i, j\}$ in Eq. (8) represents the time indices that facilitate multi-robot constraints. We assume a given criteria function $\mathbf{cr_{MR}}(v_i, v_j)$ that determines if there should be a multi-robot constraint between the two vertices $v_i$ and $v_j$. This function is conceptually similar to the indicator function used in [18], while in our previous work [10] we used a simpler criteria (relative distance between poses). The joint belief (8) can be represented by a factor graph graphical model, as illustrated in Figure 2. Different candidate paths $\mathcal{P}$ typically yield different factor graphs.

In a decentralized multi-robot framework, each robot maintains the joint belief (8) on its own while communicating to each other relevant pieces of information. We assume, for simplicity, each robot is capable of calculating the joint pdf at planning time $\mathbb{P}(X_k | \mathcal{H}_k)$ using one of the recently developed approaches (e.g. [7], [13]). We note that given transition and observation models (1), it is sufficient for each robot $r'$ to only transmit (in addition to what is required by multi-robot inference) the corresponding controls to path $\mathcal{P}^{r'}$. Any robot $r$ that receives this information can then formulate the multi-robot belief (8) [18].

Evaluating the objective function (3) for the considered paths $\mathcal{P}$ involves performing inference over the multi-robot belief (8). As shown in prior work (e.g. [6], [12]), this inference can be performed in the information space:

$$\Lambda(\mathcal{P}) = \Lambda_k + \sum_{r=1}^{R} \left[ \sum_{l=1}^{L(\mathcal{P}^r)} \Lambda_l^{r,local} + \sum_{\{i,j\}} \Lambda_{i,j}^{r,r'} \right] \quad (9)$$

where $\Lambda_l^{r,local} = (F_l^r)^T \Omega_w^r F_l^r + \sum_m (H_{l,m}^r)^T \Omega_{vlm}^r H_{l,m}^r$ and $\Lambda_{i,j}^{r,r'}$ represents the information from the multi-robot constraint term $\mathbb{P}(z_{i,j}^{r,r'} | x_{v_i}^r, x_{v_j}^{r'})$ in Eq. (8). Here, the matrices $F$ and $H$ represent appropriate Jacobians of the state transition and observation models (1), linearized about the considered candidate path and the MAP estimate of the joint state at planning (current) time. Observe that the matrices in Eq. (9) are assumed to be appropriately augmented (e.g. zero-padded) as the dimensionality of the state increases with $l$; see similar treatment e.g. in [6], [12].

Recalling that each robot $r$ has numerous candidate paths over the graph $G^r$, determining the optimal controls involves considering *all* path combinations between different robots, which is computationally intractable. Optimality here refers to choosing the best path from the set of candidate paths.

Instead, a common (sub-optimal) approach for decentralized belief space planning is for each robot $r$ to consider only its own candidate paths and the *announced* paths of other robots, see e.g. [3], [18]. The robot can then select the best path, according to the objective function (3), and announce this path to other robots, which then repeat the same procedure on their end. Such an approach reduces the exponential complexity in number of robots to a linear complexity, and can be viewed as a decentralized coordindated descent [3], [18], i.e. where robots either repeat this process until convergence [3] or at some frequency [18]. Performance guarantees of such an approach are analyzed in [3].

In particular, when an announced path of some robot $r'$ is updated (e.g. from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$), robot $r$ has to recalculate the best path by re-evaluating its candidate paths given $\mathcal{P}^{r'}_{new}$. Existing approaches perform this re-evaluation for *all* candidate paths from *scratch*. In contrast, in the following section we develop an approach to identify and efficiently re-evaluate, while re-using calculations, *only* impacted candidate paths due to an update in the announced path.

## IV. Approach

Although our approach applies for any number of robots, for simplicity we consider the case of two robots $r$ and $r'$ and re-write the objective function $J$ from Eq. (3) as $J(\mathcal{P}^r, \mathcal{P}^{r'}) = \mathbb{E}\left[\sum_{l=1}^{L}[c_l^r(b[X^r_{k+l}], u^r_{k+l}(\mathcal{P}^r)) + c_l^{r'}(b[X^{r'}_{k+l}], u^{r'}_{k+l}(\mathcal{P}^{r'}))]\right]$. In Section IV-D we then generalize back to a general number of robots.

Consider robot $r$ has already calculated belief evolution over *all* candidate paths while accounting for the announced path $\mathcal{P}^{r'}$, and the latter is now updated to $\mathcal{P}^{r'}_{new}$. The corresponding multi-robot beliefs for some candidate path $\mathcal{P}^r$ of robot $r$ are:

$$b[\mathcal{P}^r, \mathcal{P}^{r'}] = \mathbb{P}(X_k|\mathcal{H}_k)\mathbb{P}(\mathcal{P}^r|U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)) \quad (10)$$
$$\mathbb{P}(\mathcal{P}^{r'}|U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \prod_{\{i,j\}} \mathbb{P}(z^{r,r'}_{i,j}|x^r_{v_i}, x^{r'}_{v_j})$$

$$b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}] = \mathbb{P}(X_k|\mathcal{H}_k)\mathbb{P}(\mathcal{P}^r|U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)) \quad (11)$$
$$\underline{\mathbb{P}(\mathcal{P}^{r'}_{new}|U^{r'}(\mathcal{P}^{r'}_{new}), Z^{r'}(\mathcal{P}^{r'}_{new})) \prod_{\{i,j\}} \mathbb{P}(z^{r,r'}_{i,j}|x^r_{v_i}, x^{r'}_{v_j}),}$$

where the changed terms are underlined and denoted in red.

One can consider the joint beliefs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$ to be represented by appropriate two different factor graphs (see Figures 1 and 2). Re-evaluating the objective function for a candidate path $\mathcal{P}^r$ involves performing MAP inference over the updated factor graph $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$. In the general case, the factor graphs will be different for *each* candidate path $\mathcal{P}^r$.

The general concept of our approach is to track the multi-robot factors and local information *change* between the two pdfs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$. This information is then used to efficiently perform inference over the updated belief, which is required for re-evaluating the objective function.

Our approach first identifies which candidate paths $\mathcal{P}^r$ of robot $r$ are impacted as a result of the update in the announced plan, and consequently operates *only* over these paths instead of always re-calculating belief evolution over all candidate paths. Second, our approach efficiently calculates the belief evolution over these impacted paths, while re-using calculations where possible.

The main steps of the proposed approach are summarized below and described in detail in the following sections:

1) Section IV-A calculates the change in local information between $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$.
2) Section IV-B identifies the impacted candidate paths $\mathcal{P}^r$ and collects appropriate multi-robot factors to be later used for efficient belief inference.
3) Section IV-C re-evaluates the objective function for (only) the impacted candidate paths, based on the output of Sections IV-A and IV-B.

### A. Change in Local Information between $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$

We first calculate the change in local information between $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$. This calculation is used later in Alg. 2 for *consistent* inference over appropriate beliefs while avoiding double counting information that is shared by $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$. Specifically, recalling the definition (6) of a factor graph $FG_{local}(\mathcal{P}^{r'})$ that represents only the local information along path $\mathcal{P}^{r'}$ we identify which factors only appear in $FG_{local}(\mathcal{P}^{r'})$ or in $FG_{local}(\mathcal{P}^{r'}_{new})$. These factors will then be either added or removed upon re-evaluating belief evolution along impacted candidate paths $\mathcal{P}^r$. We therefore collect these factors into two separate factor graphs:

$$FG^{rmv}_{local} \doteq \{f \mid f \in FG_{local}(\mathcal{P}^{r'}) \wedge f \notin FG_{local}(\mathcal{P}^{r'}_{new})\}$$
$$FG^{add}_{local} \doteq \{f \mid f \notin FG_{local}(\mathcal{P}^{r'}) \wedge f \in FG_{local}(\mathcal{P}^{r'}_{new})\}$$

Additionally, we calculate belief evolution $b[\mathcal{P}^{r'}_{new}]$ along path $\mathcal{P}^{r'}_{new}$ taking into account *only* local information of robot $r'$, and use it to calculate the *change* in the immediate cost functions $c_l^{r'}$ between $b[\mathcal{P}^{r'}_{new}]$ and $b[\mathcal{P}^{r'}_{new}]$. Denoting this change by $\Delta c_l^{r'}$ we let $\Delta J^{r'} \doteq \mathbb{E}\left[\sum_{l=1}^{L} \Delta c_l^{r'}\right]$. This quantity will be used to very efficiently re-evaluate the objective function for candidate paths $\mathcal{P}^r$ that are *not* impacted, as discussed in Section IV-C.

### B. Impacted Paths and Change in Multi-Robot Factors

Next, we identify, among all the candidate paths of robot $r$, those paths $\mathcal{P}^r$ that are impacted as a result of the update in the announced path from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$. In other words, recalling Eqs. (10)-(11), we are interested in finding paths $\mathcal{P}^r$ such that $b[\mathcal{P}^r] \neq b'[\mathcal{P}^r]$, with

$$b[\mathcal{P}^r] = \int b[\mathcal{P}^r, \mathcal{P}^{r'}]d\mathcal{P}^{r'} \ , \ b'[\mathcal{P}^r] = \int b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]d\mathcal{P}^{r'}_{new}$$

Such paths $\mathcal{P}^r$ are marked, indicating that the objective function should be re-evaluated, a process that involves re-calculating belief evolution. On the other hand, belief evolution re-calculation is not required for candidate paths that are *not* impacted. In the latter case, the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'})$ is only updated due to the change in immediate cost functions $c_l^{r'}$ of robot $r'$, as discussed in Section IV-C.

We now describe our approach to identify the impacted paths, as well as collecting the required information that will be used in Section IV-C (Alg. 2) for efficient inference.

The *key observation* is that the belief over path $\mathcal{P}^r$ is impacted due to an announced path $\mathcal{P}^{r'}$ only if there exist multi-robot factors $\mathbb{P}(z^{r,r'}_{i,j}|x^r_{v_i}, x^{r'}_{v_j})$ or, in certain cases, if the states of robots $r$ and $r'$ are already correlated at planning time, i.e. $\mathbb{P}(X_k|\mathcal{H}_k) \neq \mathbb{P}(X_k^r|\mathcal{H}_k)\mathbb{P}(X_k^{r'}|\mathcal{H}_k)$. This is the case if, by planning time $t_k$, the robots have already performed some multi-robot update, e.g. by mutually observing a common scene.

Clearly, in absence of multi-robot factors and prior correlation, the belief over a candidate path $\mathcal{P}^r$ is not impacted by neither $\mathcal{P}^{r'}$ nor $\mathcal{P}^{r'}_{new}$. However, it is also interesting to note that also when there is prior correlation, but *no changes* in multi-robot factors between $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$, the belief over path $\mathcal{P}^r$ typically remains the same. In what follows we treat prior correlation just as a multi-robot factor, see Figure 2. However, this observation can be used to switch to a more efficient version of the algorithm that approximately recovers the pdf $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$. Investigation of this direction is left to future research.

1 **Inputs:**
2     $G^r = (V^r, E^r)$: graph of robot $r$
3     $P^{r'}, P^{r'}_{new}$: prev. and updated announced path of robot $r'$
4     $\mathbf{cr_{MR}}(v_i, v_j)$: multi-robot factor criteria function
5 **Outputs:**
6     $V^r_{inv}$: involved vertices in multi-robot factors
7     $\forall v \in V_{inv} : v.FG^{add}_{MR}, v.FG^{add}_{MR}$
8 $V^r_{inv} = \phi$ /* Initialization */
9 **foreach** $v^{r'} \in \mathcal{P}^{r'} \cup \mathcal{P}^{r'}_{new}$ **do**
10     Find all nearby vertices $\{v\} \subseteq V^r$ to $v^{r'}$ such that
11     - at least one candidate path $\mathcal{P}^r$ goes through $v$
12     - multi-robot criteria $\mathbf{cr_{MR}}(v, v^{r'})$ is satisfied
13     $V^r_{inv} = V^r_{inv} \cup \{v\}$
14     **foreach** $v_i \in \{v\}$ **do**
15         Generate multi-robot factor $f(x^r_{v_i}, x^{r'}_v)$
16         **if** $v_i \in \mathcal{P}^{r'}$ ***and*** $v_i \in \mathcal{P}^{r'}_{new}$ **then**
17             **continue**
18         **end**
19         **if** $v_i \in \mathcal{P}^{r'}$ **then**
20             Add $f(x^r_{v_i}, x^{r'}_v)$ to $v_i.FG^{rmv}_{MR}$
21         **else**
22             Add $f(x^r_{v_i}, x^{r'}_v)$ to $v_i.FG^{add}_{MR}$
23         **end**
24     **end**
25     Mark all candidate paths $\mathcal{P}^r$ that go through vertex $v_i$
26 **end**
27 **return** $V^r_{inv}$

**Algorithm 1:** `identifyInvolvedPaths`. Identify vertices $V^r_{inv} \subseteq V$ involving multi-robot factors considering announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$, and the corresponding multi-robot factors. Each vertex $v \in V_{inv}$ is associated with appropriate multi-robot factors to be later used in Alg. 2.

As mentioned in Section IV, our approach tracks the changed multi-robot factors and the local factors of robot $r'$ between the beliefs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$. This information is then used in Section IV-C to efficiently re-evaluate the belief over path $\mathcal{P}^r$. However, such a procedure is required for *each* candidate path $\mathcal{P}^r$ that has some multi-robot factors, even if several paths are identical up to some point. This would lead to the same work (i.e. computational effort) done multiple times.

To address this issue, rather than reasoning about robot $r$'s candidate paths, we reason in terms of the corresponding *vertices* in the graph $G^r$, that define the paths. Our approach, summarized in Alg. 1, considers the corresponding graph vertices and identifies the vertices $V_{inv} \subseteq V^r$ that are involved in at least one multi-robot factor due to either $\mathcal{P}^{r'}$ or $\mathcal{P}^{r'}_{new}$. See illustration in Figure 2. We then associate to each such vertex $v_i \in V_{inv}$ the *changed* multi-robot factors that involve $v_i$, i.e. any such factor $f$ should either appear in $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ or in $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$. In the former case, $f$ should be removed from the corresponding factor graph, and as such is added to $v_i.FG^{rmv}_{MR}$ (line 20); in the latter case, $f$ should be added and is thus added to $v_i.FG^{add}_{MR}$ (line 22).

Finally, the algorithm marks all paths $\mathcal{P}^r$ that include at least one vertex in $V^r_{inv}$ as impacted paths (line 25), to indicate belief re-evaluation is required.

### C. Objective Function Re-Evaluation for Candidate Paths

As mentioned in Section III, each robot $r$ evaluates the objective function by considering its candidate paths and the announced paths of different robots. Such a process requires performing inference over the belief $b[X_{k+l}]$, for each look ahead step $l$, to recover its first two moments

$$b[\mathcal{P}^r_{k+l}, \mathcal{P}^{r'}_{k+l}] = \mathcal{N}(\mu_{k+l}, \Lambda^{-1}_{k+l}), \tag{12}$$

where the general form for the information matrix $\Lambda_{k+l}$ is given by Eq. (9). Observe that if the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'})$ only includes immediate cost functions for some of the look ahead steps $l$, then the above inference is only required for these time instances. For example, one may be interested only in the uncertainty at the final step (e.g. upon reaching a goal), in which case inference should be performed only for $l = L$. On the other hand, in chance-constrained motion-planning (see e.g. [5]), belief evolution is typically needed for many (or all) look ahead steps $l$.

Since the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'})$ has been already calculated for different candidate paths $\mathcal{P}^r$ and the announced path $\mathcal{P}^{r'}$, a process that also involves inference over the corresponding beliefs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$, our objective now is to efficiently evaluate the objective function considering the *updated* announced path $\mathcal{P}^{r'}_{new}$.

Our approach for re-evaluating the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$ for each candidate path $\mathcal{P}^r$, while exploiting results from the previous inference $b[\mathcal{P}^r, \mathcal{P}^{r'}]$, is summarized in Alg. 2 and further discussed below.

The algorithm calculates the maximum a posteriori (MAP) information matrix that corresponds to the belief $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$ for each of the future time instances, which is then used for evaluating the objective function $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$. Let $\Lambda \doteq \Lambda(\mathcal{P}^r, \mathcal{P}^{r'})$ and $\Lambda' \doteq \Lambda(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$ represent the corresponding MAP information matrices to the beliefs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$, respectively. Denote also by $\Lambda_{k+l}$ the information matrix that corresponds to the belief over the first $l$ steps, $b[\mathcal{P}^r_{k+l}, \mathcal{P}^{r'}_{k+l}]$, and likewise for $\Lambda'_{k+l}$. Since inference over $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ has been already performed, the matrices $\Lambda_{k+l}$ for all steps $l$ are known. We now focus on calculating $\Lambda'_{k+l}$, for each candidate path $\mathcal{P}^r$.

1 **Inputs:**
2 $V^r_{inv}$: involved vertices in multi-robot factors
3 For each candidate path $\mathcal{P}^r$: $J(\mathcal{P}^r, \mathcal{P}^{r'})$; $\forall l \in L(\mathcal{P}^r) : \Lambda_{k+l}$ from Eq. (12)
4 $\Delta J^{r'}$ from Sec. IV-A
5 **Outputs:**
6 For each impacted candidate path $\mathcal{P}^r$: $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$; $\forall l \in L(\mathcal{P}^r) : \Lambda'_{k+l}$
7 **foreach** *candidate path* $\mathcal{P}^r$ **do**
8 **if** $\neg \mathcal{P}^r.isMarked$ **then**
9 $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new}) = J(\mathcal{P}^r, \mathcal{P}^{r'}) + \Delta J^{r'}$
10 **continue**
11 **end**
/* re-evaluate belief over $\mathcal{P}^r$ */
12 $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new}) = 0$
13 **for** $l = 1 : L(\mathcal{P}^r)$ **do**
14 **if** $\Lambda'_{k+l}$ *is not required in Eq. (3)* **then**
15 **continue**
16 **end**
/* Get previous belief $b[\mathcal{P}^r_{k+l}, \mathcal{P}^{r'}_{k+l}]$ */
17 $\Lambda_{k+l} \doteq \Lambda_{k+l}(\mathcal{P}^r, \mathcal{P}^{r'})$ from Eq. (12)
/* Initialize $\Lambda'_{k+l} \doteq \Lambda_{k+l}(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$ */
18 $\Lambda'_{k+l} = \Lambda_{k+l}$
19 **foreach** $v \in \mathcal{P}^r$ ***and*** $v \in V^r_{inv}$ **do**
/* MR factors involving $v \in V^r_{inv}$ */
20 $\Lambda'_{k+l}$ = updInfo($\Lambda'_{k+l}, v.FG^{rmv}_{MR}, l$, rmv)
21 $\Lambda'_{k+l}$ = updInfo($\Lambda'_{k+l}, v.FG^{add}_{MR}, l$, add)
22 **end**
/* Changed local info. of robot $r'$ */
23 $\Lambda'_{k+l}$ = updInfo($\Lambda'_{k+l}, FG^{add}_{local}, l$, add)
24 $\Lambda'_{k+l}$ = updInfo($\Lambda'_{k+l}, FG^{rmv}_{local}, l$, rmv)
25 Evaluate $c^r_l$ and $c^{r'}_l$ from Eq. (3)
26 **end**
27 **end**

**Algorithm 2:** evalObjFunc. Re-evaluate objective function for candidate paths $\mathcal{P}^r$ upon update in an announced path from another robot $r'$. Notations: MR=Multi-Robot; rmv = remove.

If a candidate path $\mathcal{P}^r$ has been determined in the previous section *not* to be impacted as a result of the update in the announced path (from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$), there is no need to recalculate the immediate functions $c^r_l$ of robot $r$. We note this holds true due to the considered form of $J$, where $c^r_l$ only involves $b[\mathcal{P}^r_{k+l}]$ and not also $b[\mathcal{P}^{r'}_{k+l}]$. The latter can still change due to new local information between $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$, but that change does not affect $c^r_l$ (since $b[\mathcal{P}^r] = b'[\mathcal{P}^r]$). Therefore, to get $J(\mathcal{P}^r, \mathcal{P}^{r'}_{new})$ from $J(\mathcal{P}^r, \mathcal{P}^{r'})$ we only have to update the terms $c^r_l$ (lines 8-11 in Alg. 2). This update is the same for all non-impacted paths $\mathcal{P}^r$ and is given by $\Delta J^{r'}$ from Section IV-A. We note, however, that often, $\Delta J^{r'}$ is negligible.

For each marked (impacted) path $\mathcal{P}^r$ and for each $l \in L(\mathcal{P}^r)$, we start with the previously calculated information matrix $\Lambda_{k+l}$ and update it by adding and subtracting the multi-robot and local factors that were collected as explained in Sections IV-A and IV-B. See lines 16-24 in Alg. 2.

Specifically, referring to Eq. (9), and resorting to factor graph notation $FG \doteq b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $FG' \doteq b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}]$ , the updated information matrix $\Lambda'_{k+l}$ can be written as

**Inputs:**
$FG, l$: factor graph and time index
Linearlization point = graph vertices $V$ and $\hat{X}_k$
toAddflag: indicates if to add or subtract information
$\Lambda$: input information matrix to be updated
**Outputs:**
$\Lambda$: updated information matrix
$\{f\}$ = getFactorsCausal($FG, l$)
**foreach** $f \in \{f\}$ **do**
Linearize $f$ about linearization point and calculate $\Lambda(f)$
Adjust size of $\Lambda$, if needed
**if** *toAddflag* **then**
$\Lambda = \Lambda + \Lambda(f)$
**else**
$\Lambda = \Lambda - \Lambda(f)$
**end**
**end**

**Algorithm 3:** updInfo. Update information matrix by adding or subtracting information from factors.

$$\Lambda'_{k+l} = \Lambda_{k+l} - \sum_{\substack{f \in FG \\ f \notin FG' \\ f.t \leq t_{k+l}}} \Lambda(f) + \sum_{\substack{f \in FG' \\ f \notin FG \\ f.t \leq t_{k+l}}} \Lambda(f). \quad (13)$$

The operator $f.t$ extracts the time instances involved with the factor $f$, such that the condition $f.t \leq t_{k+l}$ enforces *causality*, i.e. we do not consider factors involving states at times greater than $k + l$. The corresponding steps are summarized in Alg. 3 that is invoked by Alg. 2. We assume existence of the function getFactorsCausal that takes as input a factor graph and time $t$, and outputs only factors involving variables up to that time. Given these factors, Alg. 3 extracts the corresponding information matrices and adds or substracts these matrices as in Eq. (13). This process involves linearizing the corresponding nonlinear functions, where the linearization point is either the graph vertices $V$ or, in case states from $X_k$ are involved, the corresponding MAP estimate $\hat{X}_k$ of $\mathbb{P}(X_k|\mathcal{H}_k)$, which is known at time $k$.

We note that, similar to Eq. (9), the information matrices in Eq. (13) should be appropriately augmented: for example, the matrices $\Lambda_{k+l}$ and $\Lambda'_{k+l}$ represent uncertainty over two partially overlapping joint states $\{X^r_{k+l}(\mathcal{P}^r), X^{r'}_{k+l}(\mathcal{P}^{r'})\}$ and $\{X^r_{k+l}(\mathcal{P}^r), X^{r'}_{k+l}(\mathcal{P}^{r'}_{new})\}$, respectively.

One can go further, and perform the calculation in Eq. (13) *incrementally*, by updating $\Lambda'_{k+l+1}$ based on $\Lambda'_{k+l}$ while adding and subtracting information from appropriate factors that involve time $k + l + 1$. This would provide an efficient mechanism to evaluate the belief for each look ahead step, if that is required by the objective function $J$. We leave further investigation of this direction to future research and formulate Alg. 2 according to 'batch' version (Eq. (13)).

*Illustrative Example:* Figure 2 illustrates key aspects of our approach. The figure indicates the set $V_{inv}$ of involved vertices in multi-robot factors in either $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ by bold circle marks. As seen there are three such vertices ($v_i, v_{i+1}$ and $v_{i+2}$) and four multi-robot factors ($f_1, f_2, f_3$ and $f_4$).

As detailed in Alg. 1, each vertex $v \in V_{inv}$ includes the *changed* multi-robot factors that have to be either added or removed. In this example, for $v_i$ there are no changed factors, since although originating from different paths, $f_1$ and $f_2$ are actually identical factors. On the other hand, $v_{i+1}$ includes the factor $f_3$ to be removed, while $v_{i+2}$ includes the factor $f_4$ to be added. All the candidate paths $\mathcal{P}^r$ that go through some vertex $v \in V_{inv}$ should be updated with the multi-robot factors included in $v$.

### D. More than 2 Robots

The presented approach is not limited to 2 robots and naturally supports any number $R$ of robots, with the objective function specified in Eq. (3). In this section we briefly specify the changes in each of the algorithmic steps to accommodate this general setting.

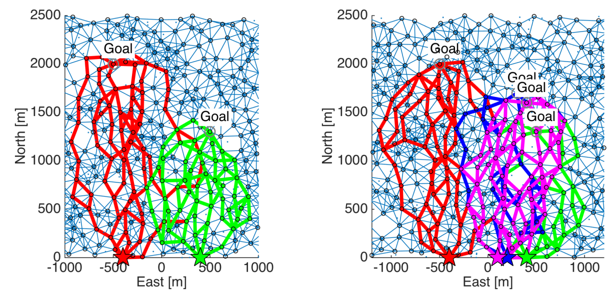
*Section IV-A:* Change in local information (Section IV-A), should be calculated with respect to all $R$ robots, excluding current robot $r$. One can go further and also incorporate within $\Delta J^{r'}$ and $\Delta J^{r''}$ the impact of changed multi-robot factors between any two robots $r'$ and $r''$. This direction is left to future research. *Section IV-B:* No modification is needed. *Section IV-C:* Algorithm 2 remains the same, however the input to the algorithm is now $J(\mathcal{P}^r, \{\mathcal{P}^{r'}\}_{r' \in \{1,\ldots,r-1,r+1,\ldots,R\}})$ instead of $J(\mathcal{P}^r, \mathcal{P}^{r'})$.

## V. Results

We demonstrate our approach in simulation considering scenario involving two and four robots operating in unknown and GPS-deprived environments that need to navigate to different goals in minimum time but also with highest accuracy. In this basic evaluation we use a prototype implementation in Matlab and GTSAM [8] to investigate key aspects of the proposed approach. The objective function (3) is $J = \sum_{r=1}^{R} \left[ \kappa^r_{goal} t^r_{goal} + \kappa^r_{\Sigma} tr\left(\Sigma^r_{goal}\right)\right]$, where $\Sigma^r_{goal}$ and $t^r_{goal}$ represent, respectively, the covariance upon reaching the goal and time of travel (or path length) for robot $r$. The parameters $\kappa^r_{goal}$ and $\kappa^r_{\Sigma}$ weight the importance of each term (we use $\kappa^r_{path} = 0.1$ and $\kappa^r_{uncert} = 1$). A probabilistic roadmap (PRM) [15] is used, to discretize the (partially unknown) environment and generate candidate paths over the roadmap. Figure 3 shows the considered scenarios for two and four robots and the generated 25 candidate paths for each robot. In this and all figures to follow, we use the notation $\star$ to indicate the starting position of each robot.

We compare our approach to a standard approach that re-evaluates from scratch belief evolution and objective function for *each* candidate path of each robot $r$ given announced paths from other robots (e.g. [3], [10], [18]). This comparison has two merits: (a) verify our approach correctly recovers the underlying pdf while identifying and re-evaluating only the impacted paths; and (b) has computational benefits.

Figure 4b shows, for the two-robot scenario, one of the candidate paths of robot $r$, an announced path of robot $r'$, and the generated multi-robot factors (cyan color); see also concept illustration in Figure 1. The corresponding belief evolution (covariance ellipses) is displayed in black. Robot




**(a)** 2 robots **(b)** 4 robots

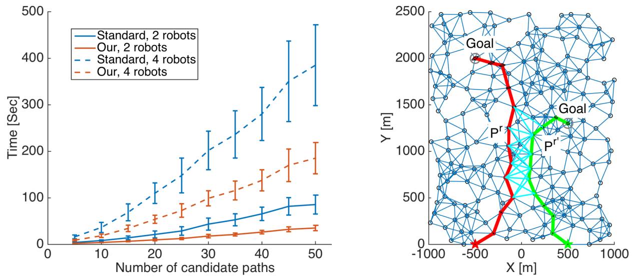
**Fig. 3:** Candidate paths shown on PRM. Robot starting positions are denoted by $\star$.

$r$ determines its best path, and announces it to other robots, which do the same; the process is repeated until convergence. Similar to [10], we use a simple heuristic for the function $\mathbf{cr_{MR}}(v_i, v_j)$ (line 4 of Alg. 1) to determine if two poses admit a multi-robot constraint: these constraints, possibly involving different future time instances, are formulated between any two poses with relative distance closer than $d = 300$ meters. More advanced methods could be implemented, e.g. considering also statistical knowledge.

The set of involved vertices in PRM, $V_{inv}$, depicted conceptually in Figure 2, is shown for robot $r$ in Figure 5 for the two-robot scenario. The figure shows marked (impacted) candidate paths of robot $r$, as a result of an update in the announced path of robot $r'$ from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$ in one of the iterations. To reduce clutter, only the impacted (marked) candidate paths of robot $r$ are shown. The corresponding multi-robot factors are color-coded: cyan indicates unchanged multi-robot factors (associated with both $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$), and yellow and magenta indicate multi-robot factors that are associated, respectively, only with $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$. These factors are appropriately then included with in the corresponding vertices in $V_{inv}$ and are used for calculating belief evolution, following Algorithms 1 and 2.

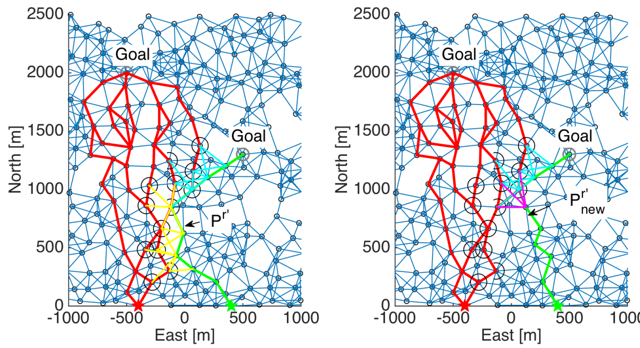
In the specific situation shown in Figure 5, only some of the candidate paths are impacted. Our approach correctly identifies, marks and consequently re-evaluates the belief over only these impacted paths. This is in contrast to the Standard approach that re-evaluates the belief from scratch over all candidate paths and recalculates the objective function for each. As a consequence, our approach exhibits substantially reduced running time, compared to the Standard approach, while producing identical results.

Figure 4a reports statistical timing results as a function of number of candidate paths $N_{cand}$ for each robot, considering the two-robot and four-robot scenarios from Figure 3. These results were obtained by running each approach 50 times, for each considered $N_{cand}$. In each such run, the scenario remains the same (goals, starting locations), while the candidate paths randomly change. As seen, as $N_{cand}$ increases the ratio between running time of the two approaches increases, in favor of our approach. In particular, for 50 candidates and two robots, our approach is 2.5 times faster compared to the standard approach (35 versus 85 seconds); A similar trend can be seen also for four robots. In all cases, identical results




**(a)** **(b)**

**Fig. 4:** (a) Statistics for running time as a function of number of candidate paths for each robot, considering groups of 2 and 4 robots. (b) Multi-robot factors (cyan color) and belief evolution (covariance ellipses) for one of the candidate paths from Figure 3, considering an announced path $\mathcal{P}^{r'}$.




**(a)** **(b)**

**Fig. 5:** Illustration of the proposed approach considering a group of two robots (see Figure 2). Vertices in $V_{inv}$ for robot $r$ given a (a) previous and (b) new announced path of robot $r'$ are shown as circles. Unchanged multi-robot factors are shown in cyan. Changed multi-robot factors associated with $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ are shown in yellow and magenta, respectively. Only impacted candidate paths of robot $r$ are shown.

were obtained, compared with the Standard approach.

## VI. Conclusions

We addressed the problem of decentralized belief space planning over high-dimensional state spaces while operating in unknown environments. Since exact solution is computationally intractable, a common approach is to address this problem within a sampling based motion planning paradigm, where each robot repeatedly considers its own candidate paths given the best paths (announced paths) transmitted by other robots. The process is typically repeated numerous times by each robot either until convergence or on a constant basis, with each time involving belief propagation along *all* candidate paths. In this paper we developed an approach that identifies and efficiently re-evaluates the belief over *only* those candidate paths that are impacted upon an update in the announced path transmitted by another robot. Determining the best path can therefore be performed without re-evaluating the utility function for each candidate path from scratch. We demonstrated in simulation our approach is capable of correctly identifying and calculating belief evolution over impacted paths, and significantly reduces computation time without any degradation in performance.

## References


[1] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato. Firm: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *Intl. J. of Robotics Research*, 2014.

[2] C. Amato, G. D. Konidaris, A. Anders, G. Cruz, J. P. How, and L. P. Kaelbling. Policy search for multi-robot coordination under uncertainty. In *Robotics: Science and Systems (RSS)*, 2015.

[3] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas. Decentralized active information acquisition: Theory and application to multi-robot slam. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.

[4] D. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[5] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 723–730, 2011.

[6] S. M. Chaves, A. Kim, and R. M. Eustice. Opportunistic sampling-based planning for active visual slam. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3073–3080. IEEE, 2014.

[7] A. Cunningham, V. Indelman, and F. Dellaert. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[8] F. Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology, 2012.

[9] G. A. Hollinger and G. S. Sukhatme. Sampling-based robotic information gathering algorithms. *Intl. J. of Robotics Research*, pages 1271–1287, 2014.

[10] V. Indelman. Towards cooperative multi-robot belief space planning in unknown environments. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, September 2015.

[11] V. Indelman. Towards multi-robot active collaborative state estimation via belief space planning. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, September 2015.

[12] V. Indelman, L. Carlone, and F. Dellaert. Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *Intl. J. of Robotics Research*, 34(7):849–882, 2015.

[13] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein. Graph-based distributed cooperative navigation for a general multi-robot measurement model. *Intl. J. of Robotics Research*, 31(9), August 2012.

[14] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Intl. J. of Robotics Research*, 30(7):846–894, 2011.

[15] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, 1996.

[16] H. Kurniawati, D. Hsu, and W. S. Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems (RSS)*, volume 2008, 2008.

[17] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *Intl. J. of Robotics Research*, 20(5):378–400, 2001.

[18] D. Levine, B. Luders, and J. P. How. Information-theoretic motion planning for constrained sensor networks. *Journal of Aerospace Information Systems*, 10(10):476–496, 2013.

[19] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. *Intl. J. of Robotics Research*, 32(7):826–840, 2013.

[20] C. Papadimitriou and J. Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[21] S. Pathak, A. Thomas, A. Feniger, and V. Indelman. Robust active perception via data-association aware belief space planning. *arXiv preprint: 1606.05124*, 2016.

[22] R. Platt, R. Tedrake, L.P. Kaelbling, and T. Lozano-Pérez. Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems (RSS)*, pages 587–593, Zaragoza, Spain, 2010.

[23] S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *Intl. J. of Robotics Research*, 2009.

[24] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Robotics: Science and Systems (RSS)*, pages 65–72, 2005.

[25] J. Van Den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *Intl. J. of Robotics Research*, 31(11):1263–1278, 2012.