



Decentralized multi-robot belief space planning in unknown environments via identification and efficient re-evaluation of impacted paths

Tal Regev¹ · Vadim Indelman²

Received: 28 November 2016 / Accepted: 7 July 2017
© Springer Science+Business Media, LLC 2017

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.

Keywords Multi-robot planning · Belief space planning · Multi-robot SLAM · Active collaborative perception

1 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 (Papadimitriou and Tsitsiklis 1987). 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. Kurniawati et al. 2008), simulation based approaches (e.g. Stachniss et al. 2005) in the context of active SLAM, sampling based approaches (e.g. Kavraki et al. 1996; LaValle and Kuffner 2001; Karaman and Frazzoli 2011) and direct trajectory optimization approaches (e.g. Platt et al. 2010; Berg et al. 2012; Indelman et al. 2015).

In particular, sampling based approaches (e.g. Prentice and Roy 2009; Bry and Roy 2011; Hollinger and Sukhatme 2014; Agha-Mohammadi et al. 2014) discretize the state space using randomized exploration strategies to

This is one of several papers published in *Autonomous Robots* comprising the Special Issue on Online Decision Making in Multi-Robot Coordination.

✉ Vadim Indelman
vadim.indelman@technion.ac.il

Tal Regev
talregev@technion.ac.il

¹ Department of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel

² Department of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel

explore the belief space in search of an optimal plan. While many of these approaches, including probabilistic roadmap (PRM) (Kavraki et al. 1996), rapidly exploring random trees (RRT) (LaValle and Kuffner 2001), and RRT* and Rapidly-exploring Random Graph (RRG) (Karaman and Frazzoli 2011), 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) (Prentice and Roy 2009) and the rapidly-exploring random belief trees (RRBT) (Bry and Roy 2011), 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. Hollinger and Sukhatme 2014).

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. Stachniss et al. (2005), Chaves et al. (2014), Indelman et al. (2015). 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. Levine et al. 2013; Atanasov et al. 2015; Indelman 2015a, b). In particular, in a recent work (Indelman 2015a) 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, which has numerous advantages compared to the centralized case (e.g. robustness to failure, communication to a single computational unit is not required).

Unfortunately, solving exactly the corresponding decentralized POMDP problem is computationally intractable and has been shown to be nondeterministic exponential (NEXP) complete (Bernstein et al. 2002), 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. Amato et al. 2016).

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. Levine et al. 2013; Atanasov et al. 2015). 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 Atanasov et al. (2015).

Yet, existing methods calculate the belief evolution over *all* candidate paths from *scratch* each time a new announced plan from another robot is received, (see, e.g. Levine et al. 2013; Atanasov et al. 2015), which by itself can be computationally extensive operation. Another related recent body of work (Agha-Mohammadi et al. 2014; Agha-mohammadi et al. 2015) proposes to dynamically re-plan in belief space to address discrepancy between the actual motion and observation models and those used in the planning stage. However, as opposed to the research presented herein, these works do not focus on a multi-robot setup and do not consider unknown or uncertain environments.

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 Fig. 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.

The present paper is an extension of the work presented in (Regev and Indelman 2016). As a further contribution, in this manuscript we extend our approach to also support scenarios where states of different robots are correlated at planning time (Sect. 5), for example, due to previous mutual landmark observations. Moreover, we provide an extensive experimental evaluation in simulation (Sect. 7.2), where the robots have to accurately autonomously navigate to multiple pre-defined goals in unknown environments. The considered scenario involves multi-robot SLAM and multiple planning sessions in some of which the robots' states are correlated.

The remainder of this paper is organized as follows. Section 2 introduces notations and formulates the addressed problem. Section 3 presents decentralized sampling based planning framework that is used as a baseline method. Section 4 presents our multi-robot belief space planning approach. Section 5 focuses on the case where the states of different robots are correlated at planning time. Section 7 presents experimental results, considering several simulated scenarios. Conclusions are provided in Sect. 8.

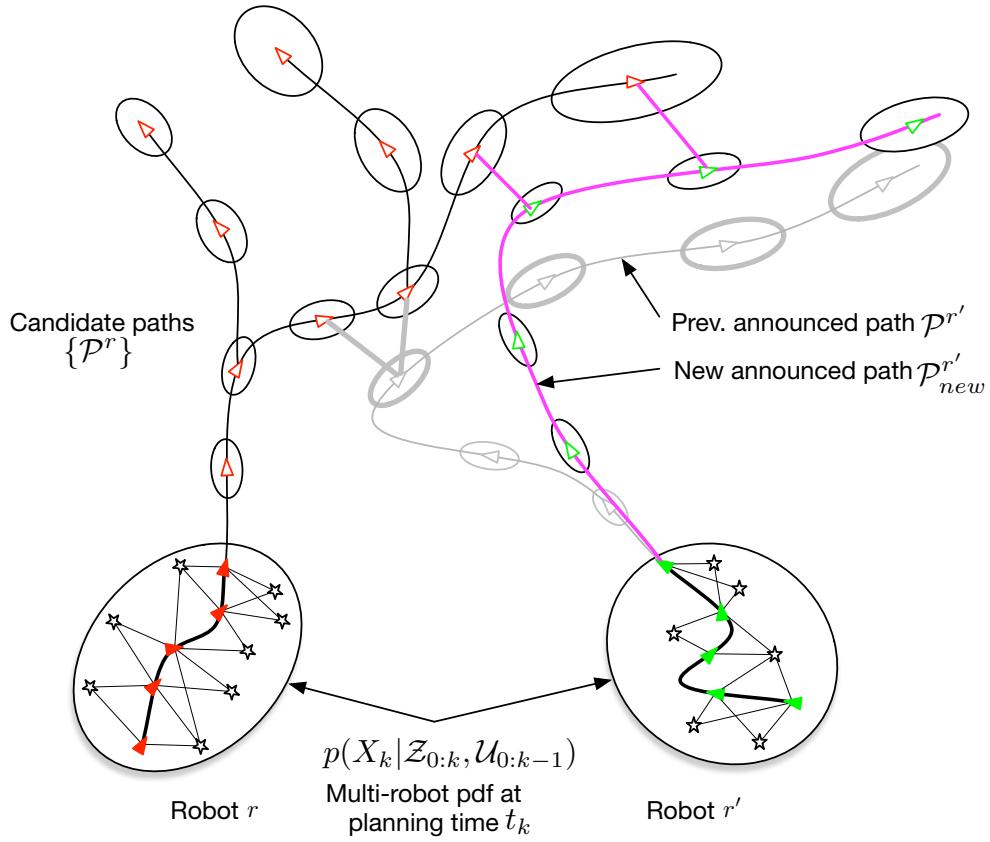


Fig. 1 Illustration of the proposed concept. The figure shows belief evolution over a few candidate paths of robot \$r\$ given an announced path \$\mathcal{P}^{r'}\$ from robot \$r'\$ and the corresponding multi-robot constraints that can represent, e.g., future mutual observations of environments unknown at planning time (Indelman 2015a). Upon an update in an

announced path from \$\mathcal{P}^{r'}\$ to \$\mathcal{P}_{new}^{r'}\$, 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 (Color figure online)

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

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

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

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

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,

$$\mathbb{P}(Z_i^r | X_i^r) = \prod_{j=1}^{n_i} \mathbb{P}(z_{i,j}^r | X_{i,j}^r). \quad (3)$$

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. Olson and Agarwal 2013) and, recently, in belief space planning (Pathak et al. 2016).

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-1}^r\}_{r=1}^R\$.

Let J denote a user-defined objective function

$$J(\mathcal{U}) = \mathbb{E} \left[\sum_{l=1}^L c_l(b[X_{k+l}], u_{k+l}) \right], \quad (4)$$

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:

$$J(\mathcal{U}) = \mathbb{E} \left[\sum_{l=1}^L \sum_{r=1}^R c_l^r(b[X_{k+l}^r], u_{k+l}^r) \right], \quad (5)$$

where $b[X_{k+l}^r] = \int_{-X_{k+l}^r} b[X_{k+l}]$ 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. Indelman 2015b).

In this paper, our objective is to find the optimal controls

$$\mathcal{U}^* = \arg \min_{\mathcal{U}} J(\mathcal{U}) \quad (6)$$

for all robots in the group, considering a multi-robot decentralized framework discussed below.

3 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 $\mathcal{P}^r \doteq \{v_0, \dots, v\}$ 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

$$b[\mathcal{P}^r] \doteq \mathbb{P}(X_k^r, x_{v_0}^r, \dots, x_v^r | \mathcal{H}_k^r, U(\mathcal{P}^r), Z(\mathcal{P}^r)), \quad (7)$$

where $U(\mathcal{P}^r)$ and $Z(\mathcal{P}^r)$ 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))

$$b[\mathcal{P}^r] = \mathbb{P}(X_k^r | \mathcal{H}_k^r) \mathbb{P}(\mathcal{P}^r | U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)), \quad (8)$$

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

$$FG_{local}(\mathcal{P}^r) \doteq \prod_{l=1}^{L(\mathcal{P}^r)} \mathbb{P}(x_{v_l}^r | x_{v_{l-1}}^r, u_{v_{l-1}}^r) \mathbb{P}(Z_{v_l}^r | X_{k+l}^r). \quad (9)$$

Throughout the paper we will often use the factor graph graphical model to represent a pdf. The factor graph for the pdf from Eq. (9) 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.:

$$X_{k+l}^r = X_{k+l}^r(\mathcal{P}^r) \equiv \mathcal{P}_{k+l}^r \doteq \{X_k^r, x_{v_0}^r, \dots, x_{v_l}^r\}. \quad (10)$$

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

$$b[\mathcal{P}] = \mathbb{P}(X_k | \mathcal{H}_k) \prod_{r=1}^R \left[\prod_{l=1}^{L(\mathcal{P}^r)} \mathbb{P}(x_{v_l}^r | x_{v_{l-1}}^r, u_{v_{l-1}}^r) \cdot \mathbb{P}(Z_{v_l}^r | X_{k+l}^r) \prod_{\{i, j\}} \mathbb{P}(z_{i,j}^{r,r'} | x_{v_i}^r, x_{v_j}^{r'}) \right], \quad (11)$$

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 (Indelman 2015a).

The index set $\{i, j\}$ in Eq. (11) represents the time indices that facilitate multi-robot constraints. We assume a given criteria function $\text{cr}_{\text{MR}}(v_i, v_j)$ 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 Levine et al. (2013), while in our previous work (Indelman 2015a) we used a simpler criteria (relative distance between poses). The joint belief (11) can be represented by a factor graph graphical model, as illustrated in Fig. 2. Different candidate paths \mathcal{P} typically yield different factor graphs.

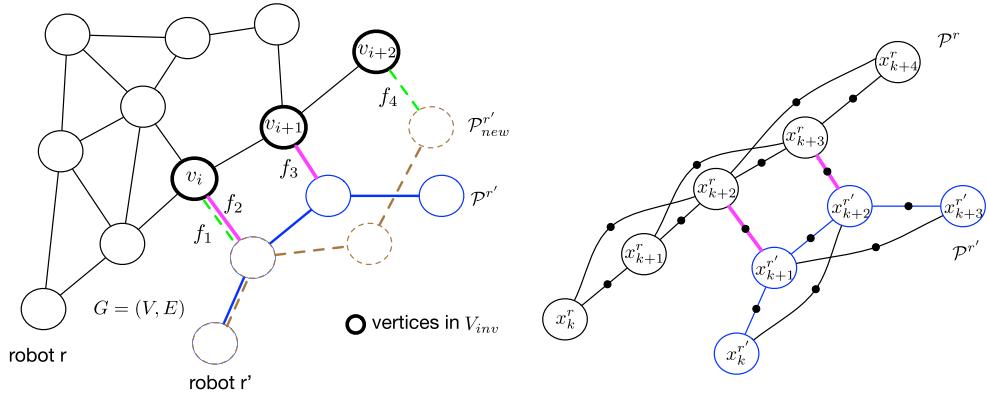


Fig. 2 (left) Graph $G = (V, E)$ along which different candidate paths \mathcal{P}^r of robot r can be defined. Announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$ from robot r' facilitate multi-robot factors f_1, f_2, f_3 and f_4 . (right) An example

In a decentralized multi-robot framework, each robot maintains the joint belief (11) 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. Cunningham et al. 2013; Indelman et al. 2012). 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 (11) (Levine et al. 2013).

Evaluating the objective function (5) for the considered paths \mathcal{P} involves performing inference over the multi-robot belief (11). As shown in prior work (e.g. Indelman et al. 2015; Chaves et al. 2014), this inference can be performed in the information space:

$$\Lambda(\mathcal{P}) = \Lambda_k + \sum_{r=1}^R \left[\sum_{l=1}^{L(\mathcal{P}^r)} \Lambda_l^{r,local} + \sum_{\{i,j\}} \Lambda_{i,j}^{r,r'} \right], \quad (12)$$

where

$$\Lambda_l^{r,local} = (F_l^r)^T \Omega_w^r F_l^r + \sum_m (H_{l,m}^r)^T \Omega_{vlm}^r H_{l,m}^r, \quad (13)$$

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. (11). 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. (12) 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 Indelman et al. (2015), Chaves et al. (2014).

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 $\mathcal{P}_{new}^{r'}$

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. Levine et al. (2013), Atanasov et al. (2015). The robot can then select the best path, according to the objective function (5), 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 coordinated descent (Levine et al. 2013; Atanasov et al. 2015), i.e. where robots either repeat this process until convergence (Atanasov et al. 2015) or at some frequency (Levine et al. 2013). Performance guarantees of such an approach are analyzed in Atanasov et al. (2015).

In particular, when an announced path of some robot r' is updated (e.g. from $\mathcal{P}^{r'}$ to $\mathcal{P}_{new}^{r'}$), robot r has to recalculate the best path by re-evaluating its candidate paths given $\mathcal{P}_{new}^{r'}$. 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.

4 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. (5) as

$$J(\mathcal{P}^r, \mathcal{P}^{r'}) = \mathbb{E} \left[\sum_{l=1}^L \left[c_l^r(b[X_{k+l}^r], u_{k+l}^r(\mathcal{P}^r)) + c_l^{r'}(b[X_{k+l}^{r'}], u_{k+l}^{r'}(\mathcal{P}^{r'})) \right] \right]. \quad (14)$$

In Sect. 4.4 we consider again 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 $\mathcal{P}_{new}^{r'}$. The corresponding multi-robot beliefs for some candidate path \mathcal{P}^r of robot r are:

$$b[\mathcal{P}^r, \mathcal{P}^{r'}] = \mathbb{P}(X_k | \mathcal{H}_k) \mathbb{P}(\mathcal{P}^r | U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)) \mathbb{P}(\mathcal{P}^{r'} | U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \prod_{\{i,j\}} \mathbb{P}(z_{i,j}^{r,r'} | x_{v_i}^r, x_{v_j}^{r'}) \quad (15)$$

$$b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}] = \mathbb{P}(X_k | \mathcal{H}_k) \mathbb{P}(\mathcal{P}^r | U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)) \mathbb{P}(\mathcal{P}_{new}^{r'} | U^{r'}(\mathcal{P}_{new}^{r'}), Z^{r'}(\mathcal{P}_{new}^{r'})) \prod_{\{i,j\}} \mathbb{P}(z_{i,j}^{r,r'} | x_{v_i}^r, x_{v_j}^{r'}), \quad (16)$$

where the changed terms are underlined.

One can consider the joint beliefs $b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$ to be represented by appropriate two different factor graphs (see Figs. 1, 2). Re-evaluating the objective function for a candidate path \mathcal{P}^r involves performing MAP inference over the updated factor graph $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$. 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 $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$. 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 4.1 calculates the change in local information between $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$.
2. Section 4.2 identifies the impacted candidate paths \mathcal{P}^r and collects appropriate multi-robot factors to be later used for efficient belief inference.
3. Section 4.3 re-evaluates the objective function for (only) the impacted candidate paths, based on the output of Sects. 4.1 and 4.2.

4.1 Change in local information between $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$

We first calculate the change in local information between $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$. This calculation is used later in Algorithm 2 for *consistent* inference over appropriate beliefs while avoiding double counting information that is shared by $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$. Specifically, recalling the definition (9) 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 $FG_{local}(\mathcal{P}_{new}^{r'})$. 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:

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

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

Additionally, we calculate belief evolution $b[\mathcal{P}_{new}^{r'}]$ along path $\mathcal{P}_{new}^{r'}$ 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 $b[\mathcal{P}_{new}^{r'}]$ and $b[\mathcal{P}^{r'}]$. Denoting this change by $\Delta c_l^{r'}$ we let

$$\Delta J^{r'} \doteq \mathbb{E} \left[\sum_{l=1}^L \Delta c_l^{r'} \right]. \quad (17)$$

For example, if $c_l^{r'}$ quantifies uncertainty, e.g. $c_l^{r'} = \det(\Lambda_{k+l}^{r'})$, then $\Delta J^{r'} = \det(\Lambda_{k+l}^{r'}) - \det(\Lambda_{k+l,new}^{r'})$ where $\Lambda_{k+l}^{r'}$ and $\Lambda_{k+l,new}^{r'}$ are obtained, respectively, from $b[\mathcal{P}^{r'}]$ and $b[\mathcal{P}_{new}^{r'}]$ at the l th step along appropriate paths (i.e. $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$). Note that the expectation operator in such a case can be dropped as the posterior information (covariance) matrix can be typically approximated well by a single Gauss Newton iteration, and hence it is not a function of future observations (see e.g. Indelman et al. 2015).

The term $\Delta J^{r'}$ from Eq. (17) will be used to very efficiently re-evaluate the objective function for candidate paths \mathcal{P}^r that are *not* impacted, as discussed in Sect. 4.3.

4.2 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 $\mathcal{P}_{new}^{r'}$. In other words, recalling Eqs. (15, 16), we are interested in finding paths \mathcal{P}^r such that $b[\mathcal{P}^r] \neq b'[\mathcal{P}^r]$, with

$$b[\mathcal{P}^r] = \int b[\mathcal{P}^r, \mathcal{P}^{r'}] d\mathcal{P}^{r'}, \quad b'[\mathcal{P}^r] = \int b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}] d\mathcal{P}_{new}^{r'} \quad (18)$$

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' of robot r' , as discussed in Sect. 4.3.

We now describe our approach to identify the impacted paths, as well as collecting the required information that will be used in Sect. 4.3 (Algorithm 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 $\mathbb{P}(z_{i,j}^{r,r'} | x_{v_i}^r, x_{v_j}^{r'})$ or, in certain cases, if the states of robots r and r' are already correlated at planning time, i.e.

$$\mathbb{P}(X_k | \mathcal{H}_k) \neq \mathbb{P}(X_k^r | \mathcal{H}_k)\mathbb{P}(X_k^{r'} | \mathcal{H}_k). \quad (19)$$

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. We refer to this case (i.e. Eq. (19) holds) as *prior correlation*.

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 $\mathcal{P}_{new}^{r'}$. 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 $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$, the belief over path \mathcal{P}^r typically remains the same. We defer further discussion regarding prior correlation to Sect. 5, and in what follows we treat prior correlation between states x_k^r and $x_k^{r'}$, if such correlation exists, as a multi-robot factor, see Fig. 2.

As mentioned in Sect. 4, 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 $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$. This information is then used in Sect. 4.3 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 Algorithm 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 $\mathcal{P}_{new}^{r'}$. See illustration in Fig. 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 $b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$. In the former case, f should be removed from the corresponding factor graph, and as such

```

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

```

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

is added to $v_i.FG_{MR}^{rmv}$ (line 22); in the latter case, f should be added and is thus added to $v_i.FG_{MR}^{add}$ (line 24).

Finally, the algorithm marks all paths \mathcal{P}^r that include at least one vertex in V_{inv}^r as impacted paths (line 27), to indicate belief re-evaluation is required.

4.3 Objective function re-evaluation for candidate paths

As mentioned in Sect. 3, 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

$$b[\mathcal{P}_{k+l}^r, \mathcal{P}_{k+l}^{r'}] = \mathcal{N}(\mu_{k+l}, \Lambda_{k+l}^{-1}), \quad (20)$$

where the general form for the information matrix Λ_{k+l} is given by Eq. (12). 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. Bry and Roy 2011), 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 $\mathcal{P}_{new}^{r'}$.

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

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

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 $\mathcal{P}_{new}^{r'}$), there is no need to recalculate the immediate functions c_l^r of robot r . We note this holds true due to the considered form of J , where c_l^r only involves $b[\mathcal{P}_{k+l}^r]$ and not also $b[\mathcal{P}_{k+l}^{r'}]$. The latter can still change due to new local information between $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$, but that change does not affect c_l^r (since $b[\mathcal{P}^r] = b'[\mathcal{P}^r]$). Therefore, to get $J(\mathcal{P}^r, \mathcal{P}_{new}^{r'})$ from $J(\mathcal{P}^r, \mathcal{P}^{r'})$ we only have to update the terms $c_l^{r'}$ (lines 10–13 in Algorithm 2). This update is the same for all non-impacted paths \mathcal{P}^r and is given by $\Delta J^{r'}$ from Sect. 4.1. 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 Λ_{k+l} and update it by adding and subtracting the multi-robot and local factors that were collected as explained in Sects. 4.1 and 4.2. See lines 18–26 in Algorithm 2.

Specifically, referring to Eq. (12), and resorting to factor graph notation $FG \doteq b[\mathcal{P}^r, \mathcal{P}^{r'}]$ and $FG' \doteq b[\mathcal{P}^r, \mathcal{P}_{new}^{r'}]$,

```

1 Inputs:
2   |  $V_{inv}^r$ : involved vertices in multi-robot factors
3   | For each candidate path  $\mathcal{P}^r$ :  $J(\mathcal{P}^r, \mathcal{P}^{r'})$ ;  $\forall l \in L(\mathcal{P}^r) : \Lambda_{k+l}$ 
4   | from Eq. (20)
5   |  $\Delta J^{r'}$  from Sect. 4.1
6 Outputs:
7   | For each impacted candidate path  $\mathcal{P}^r$ :  $J(\mathcal{P}^r, \mathcal{P}_{new}^{r'})$ ;
8   |  $\forall l \in L(\mathcal{P}^r) : \Lambda'_{k+l}$ 
9 foreach candidate path  $\mathcal{P}^r$  do
10  | if  $\neg \mathcal{P}^r.isMarked$  then
11    |    $J(\mathcal{P}^r, \mathcal{P}_{new}^{r'}) = J(\mathcal{P}^r, \mathcal{P}^{r'}) + \Delta J^{r'}$ 
12    |   continue
13  | end
14  | /* re-evaluate belief over  $\mathcal{P}^r$  */ *
15  |  $J(\mathcal{P}^r, \mathcal{P}_{new}^{r'}) = 0$ 
16  | for  $l = 1 : L(\mathcal{P}^r)$  do
17    |   if  $\Lambda'_{k+l}$  is not required in Eq. (5) then
18      |       | continue
19  |   end
20  |   /* Get previous belief  $b[\mathcal{P}_{k+l}^r, \mathcal{P}_{k+l}^{r'}]$  */ *
21  |    $\Lambda_{k+l} \doteq \Lambda_{k+l}(\mathcal{P}^r, \mathcal{P}^{r'})$  from Eq. (20)
22  |   /* Initialize  $\Lambda'_{k+l} \doteq \Lambda_{k+l}(\mathcal{P}^r, \mathcal{P}_{new}^{r'})$  */ *
23  |    $\Lambda'_{k+l} = \Lambda_{k+l}$ 
24  |   foreach  $v \in \mathcal{P}^r$  and  $v \in V_{inv}^r$  do
25    |     | /* MR factors involving  $v \in V_{inv}^r$  */ *
26    |     |  $\Lambda'_{k+l} = updInfo(\Lambda'_{k+l}, v.FG_{MR}^{rmv}, l, rmv)$ 
27    |     |  $\Lambda'_{k+l} = updInfo(\Lambda'_{k+l}, v.FG_{MR}^{add}, l, add)$ 
28  |   end
29  |   /* Changed local info. of robot  $r'$  */ *
30  |    $\Lambda'_{k+l} = updInfo(\Lambda'_{k+l}, FG_{local}^{add}, l, add)$ 
31  |    $\Lambda'_{k+l} = updInfo(\Lambda'_{k+l}, FG_{local}^{rmv}, l, rmv)$ 
32  |   Evaluate  $c_l^r$  and  $c_l^{r'}$  from Eq. (5)
33  | end
34 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

the updated information matrix Λ'_{k+l} can be written as

$$\Lambda'_{k+l} = \Lambda_{k+l} - \sum_{\substack{f \in FG \\ f \notin FG' \\ f.t \leq t_{k+l}}} \Lambda(f) + \sum_{\substack{f \in FG' \\ f \notin FG \\ f.t \leq t_{k+l}}} \Lambda(f). \quad (21)$$

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 sum-

```

1 Inputs:
2   |  $FG, l$ : factor graph and time index
3   | Linearization point = graph vertices  $V$  and  $\hat{X}_k$ 
4   | toAddflag: indicates if to add or subtract information
5   |  $\Lambda$ : input information matrix to be updated
6
7 Outputs:
8   |  $\Lambda$ : updated information matrix
9
10  $\{f\} = \text{getFactorsCausal}(FG, l)$ 
11 foreach  $f \in \{f\}$  do
12   | Linearize  $f$  about linearization point and calculate  $\Lambda(f)$ 
13   | Adjust size of  $\Lambda$ , if needed
14   | if toAddflag then
15   |   |  $\Lambda = \Lambda + \Lambda(f)$ 
16   | else
17   |   |  $\Lambda = \Lambda - \Lambda(f)$ 
18   | end
19 end

```

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

marized in Algorithm 3 that is invoked by Algorithm 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, Algorithm 3 extracts the corresponding information matrices and adds or subtracts these matrices as in Eq. (21). 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. (12), the information matrices in Eq. (21) should be appropriately augmented: for example, the matrices Λ_{k+l} and Λ'_{k+l} represent uncertainty over two partially overlapping joint states $\{X_{k+l}^r(\mathcal{P}^r), X_{k+l}^{r'}(\mathcal{P}^{r'})\}$ and $\{X_{k+l}^r(\mathcal{P}^r), X_{k+l}^{r'}(\mathcal{P}_{new}^{r'})\}$, respectively.

One can go further, and perform the calculation in Eq. (21) *incrementally*, by updating Λ'_{k+l+1} based on Λ'_{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 Algorithm 2 according to ‘batch’ version (Eq. (21)).

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 $\mathcal{P}_{new}^{r'}$ 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 Algorithm 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 .

Remark Many real world scenarios require obstacle detection and avoidance. In such a case, a detection of a previously unmapped, potentially dynamic, obstacle can trigger a re-planning session, for example, as done in our previous work (Indelman 2017). A straightforward application of the method proposed herein would improve efficiency, while treating each of the re-planning sessions separately. However, we envision similar reasoning, with appropriate adaptation, could be used also in-between re-planning sessions. This aspect is outside the scope of the current work and is left to future research.

4.4 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. (5). In this section we briefly specify the changes in each of the algorithmic steps to accommodate this general setting.

Section 4.1 Change in local information, 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 4.2 No modification is needed. Section 4.3 Algorithm 2 remains the same, however the input to the algorithm is now $J(\mathcal{P}^r, \{\mathcal{P}^{r'}\}_{r' \in \{1, \dots, r-1, r+1, \dots, R\}})$ instead of $J(\mathcal{P}^r, \mathcal{P}^{r'})$.

Remark For more than two robots, the efficiency of the proposed approach could vary for different orders of announced updated robot paths. However, our claim is that given an arbitrary such order, the proposed approach improves efficiency compared to the state of the art announced path approach. We further analyze computational complexity aspects in Sect. 6.

5 Prior correlation

In this section we revisit the case where at planning time k , robot states are already correlated, e.g. due to observation of a common scene, see illustration in Fig. 3. In other words Eq. (19) holds:

$$\mathbb{P}(X_k|\mathcal{H}_k) \neq \mathbb{P}(X_k^r|\mathcal{H}_k)\mathbb{P}(X_k^{r'}|\mathcal{H}_k). \quad (22)$$

As will be seen, our approach is applicable also in such a case with minor changes. Thus, the proposed approach

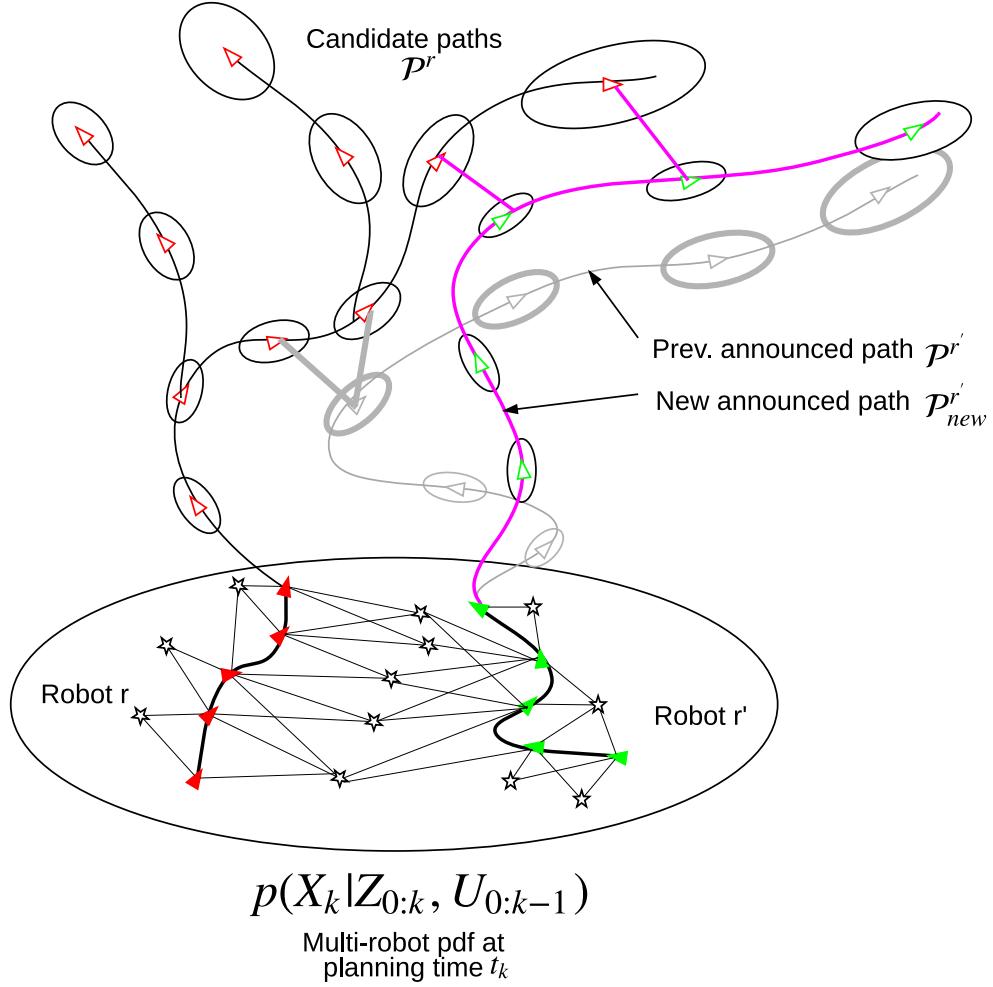


Fig. 3 Illustration of the proposed concept with correlation between robot states at planning time k (referred also as prior correlation in the text) due to mutual past landmark observations (compare with Fig. 1)

also supports more realistic, unrolling scenarios where robot states become correlated at some point and the candidate paths can go through unknown or previously mapped areas, or a combination of both.

Prior correlation at planning time k can be expressed as

$$\mathbb{P}(x_k^r, x_k^{r'} | \mathcal{H}_k) = \int_{-\infty}^{\infty} \mathbb{P}(X_k | \mathcal{H}_k) \doteq \mathcal{N}(\cdot, \Sigma(x_k^r, x_k^{r'})), \quad (23)$$

where \cdot denotes some entry that is not of interest in the current context, and

$$\Sigma(x_k^r, x_k^{r'}) \doteq \begin{bmatrix} \Sigma_{x_k^r, x_k^r} & \Sigma_{x_k^r, x_k^{r'}} \\ \Sigma_{x_k^{r'}, x_k^r} & \Sigma_{x_k^{r'}, x_k^{r'}} \end{bmatrix}. \quad (24)$$

The correlation (or cross-covariance) term $\Sigma_{x_k^r, x_k^{r'}}$ will be non-zero because of Eq. (22). Conceptually, marginalizing past robot poses and observed landmarks, the pdf (23)

induces a multi-robot factor between the variables x_k^r and $x_k^{r'}$, as depicted in Fig. 4.

Now, considering some candidate path \mathcal{P}^r of robot r and previous and new announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$ from robot r' , there are two possible cases (see Fig. 4): **a** there are some multi-robot factors between \mathcal{P}^r and $\mathcal{P}^{r'}$, and/or between \mathcal{P}^r and $\mathcal{P}_{new}^{r'}$; or **b** there are no multi-robot factors between \mathcal{P}^r and $\mathcal{P}^{r'}$ and also no multi-robot factors between \mathcal{P}^r and $\mathcal{P}_{new}^{r'}$.

In the first case, prior correlation can be considered just as an additional multi-robot factor that is treated similarly to other multi-robot factors by our approach (see Sect. 4 and Fig. 4a and b).

The second case (no multi-robot factors but with prior correlation) deserves further analysis. Figure 4c and d show a diagram of such a scenario. Since there are no multi-robot factors, the posterior beliefs from Eqs. (15) and (16) turn into

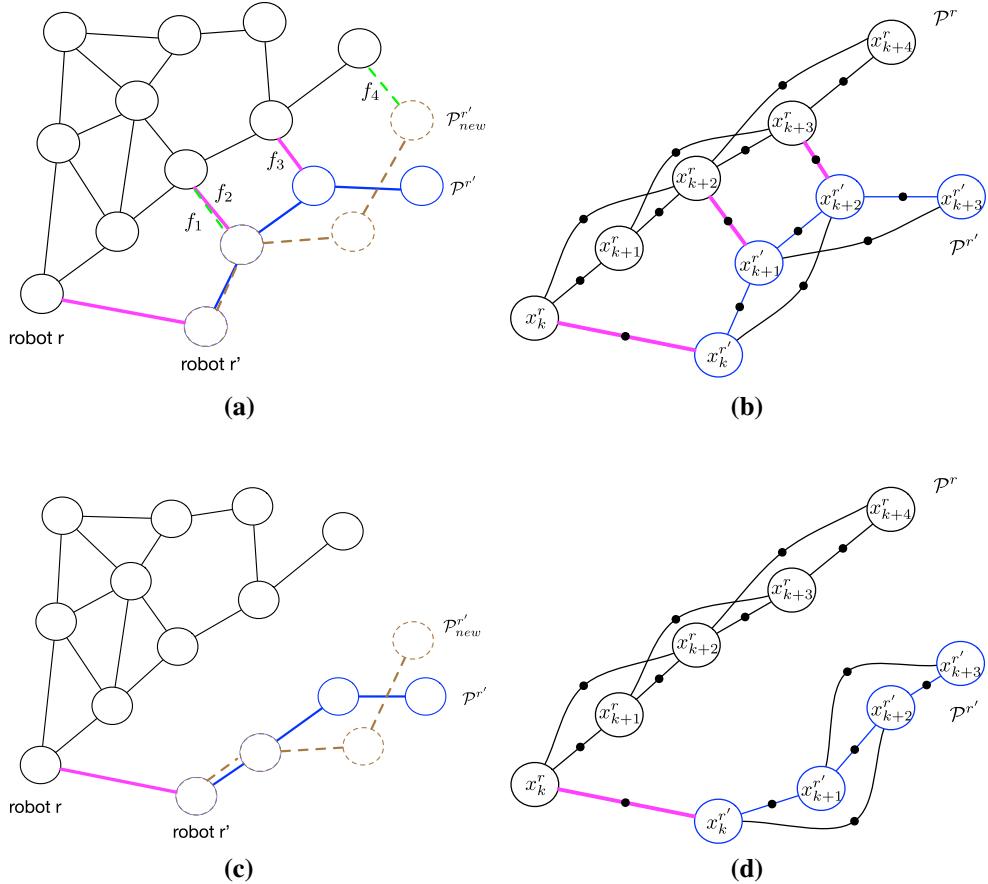


Fig. 4 Prior correlation with and without multi-robot factors: **a** and **c** show graph $G = (V, E)$ along with different candidate paths \mathcal{P}^r of robot r , and previous and new announced paths by robot r' (\mathcal{P}'^r and $\mathcal{P}'^{r'new}$). **a** Scenario with prior correlation and multi-robot factors;

b Corresponding factor graph for paths \mathcal{P}^r and \mathcal{P}'^r ; **c** Scenario with prior correlation but without multi-robot factors; **d** Corresponding factor graph for paths \mathcal{P}^r and \mathcal{P}'^r . Compare to Fig. 2 where prior correlation is not considered

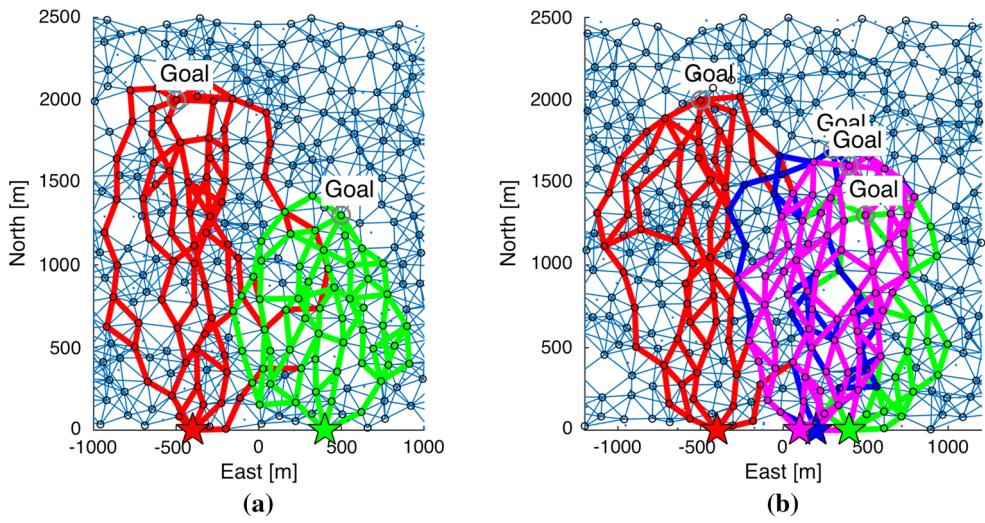


Fig. 5 Candidate paths shown on PRM. Robot starting positions are denoted by \star

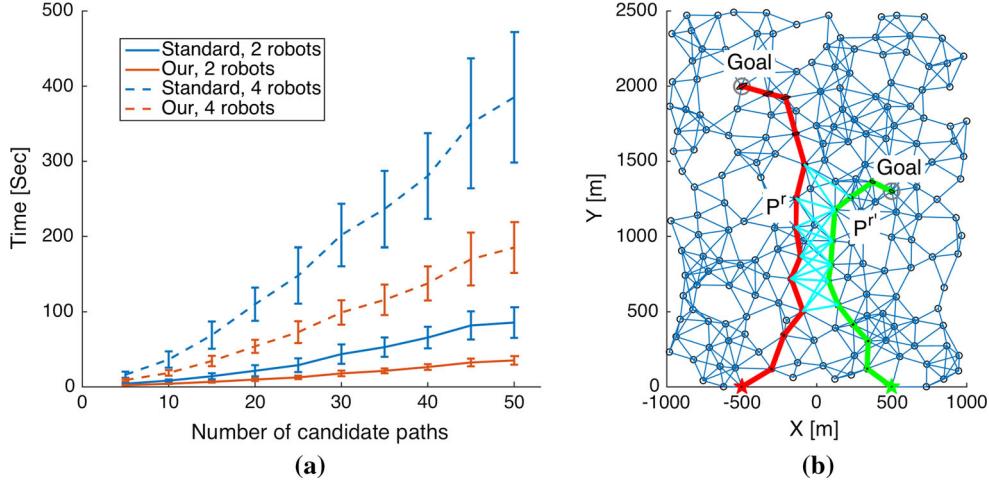


Fig. 6 **a** Statistics for running time as a function of number of candidate paths for each robot, considering groups of 2 and 4 robots. **b** Multi-robot factors (cyan color) and belief evolution (covariance ellipses) for one of the candidate paths from Fig. 5, considering an announced path P' (Color figure online)

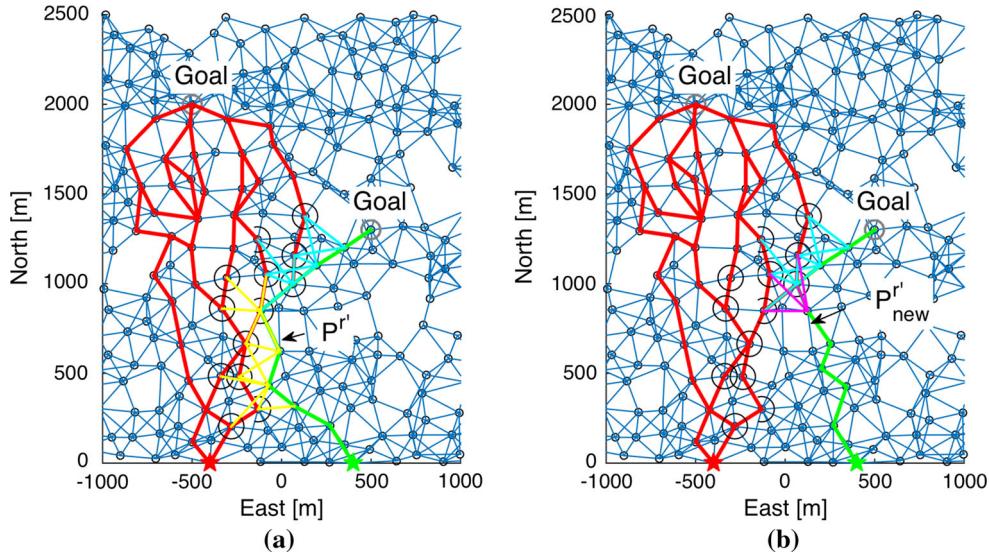


Fig. 7 Illustration of the proposed approach considering a group of two robots (see Fig. 2). Vertices in V_{inv} for robot r given a **a** previous and **b** new announced path of robot r' are shown as circles. Unchanged multi-

robot factors are shown in cyan. Changed multi-robot factors associated with P'^r and $P'^{r'}_{new}$ are shown in yellow and magenta, respectively. Only impacted candidate paths of robot r are shown (Color figure online)

$$\begin{aligned} b[\mathcal{P}^r, \mathcal{P}^{r'}] &= \mathbb{P}(X_k | \mathcal{H}_k) \mathbb{P}(\mathcal{P}^r | U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)) \\ &\quad \mathbb{P}(\mathcal{P}^{r'} | U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \\ b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}] &= \mathbb{P}(X_k | \mathcal{H}_k) \mathbb{P}(\mathcal{P}^r | U^r(\mathcal{P}^r), Z^r(\mathcal{P}^r)) \\ &\quad \mathbb{P}(\mathcal{P}^{r'}_{new} | U^{r'}(\mathcal{P}^{r'}_{new}), Z^{r'}(\mathcal{P}^{r'}_{new})), \end{aligned}$$

where the changed entries are underlined.

In the above case, if at planning time there was no prior correlation (i.e. x_k^r and $x_k^{r'}$ are not correlated), as considered in Sect. 4.2, then \mathcal{P}^r would not be impacted by the change in the announced path from $\mathcal{P}^{r'}$ to $\mathcal{P}^{r'}_{new}$. In other words (see also Eq. (18)):

$$b[\mathcal{P}^r] \doteq \int b[\mathcal{P}^r, \mathcal{P}^{r'}] d\mathcal{P}^{r'} \equiv b'[\mathcal{P}^r] \doteq \int b[\mathcal{P}^r, \mathcal{P}^{r'}_{new}] d\mathcal{P}^{r'}_{new}, \quad (25)$$

and thus, \mathcal{P}^r would not be marked. However, this does not hold in general in the presence of prior correlation.

Interestingly, however, despite having prior correlation, we observe that as long as the paths \mathcal{P}^r , $\mathcal{P}^{r'}$ and $\mathcal{P}^{r'}_{new}$ go through unknown areas with no sources of absolute information (such as GPS or known landmarks), any such candidate

Table 1 Large-scale scenarios description

Scenario	Description	Big covariance	MR factors
Scenario1	All robots have small and identical covariances at the beginning. MR factors are used within planning	No	Yes
Scenario2	All robots have small and identical covariances at the beginning. MR factors are <i>not</i> used within planning	No	No
Scenario3	Red and green robots have large uncertainty covariances at the beginning. All robots use MR factors within planning	Yes	Yes

path \mathcal{P}^r of robot r is *not* impacted due to change in the announced path (from $\mathcal{P}^{r'}$ to $\mathcal{P}_{new}^{r'}$) and thus should *not* be marked, thereby saving calculations.

We now illustrate this observation in a simple example, and then discuss in Sect. 5.2 the more general case, where the above conditions are not met, and discuss a slight modification to our approach.

5.1 Simple example

We consider a simple example where paths $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$ only include a single look ahead step. Both paths pass through unknown areas and thus we assume existence only of visual odometry measurements z^{VO} that provide relative information between consecutive states. The posterior over $x_k^{r'}$ and $x_{k+1}^{r'}$, given z^{VO} from either $\mathcal{P}^{r'}$ or $\mathcal{P}_{new}^{r'}$ is

$$\mathbb{P}(x_k^{r'}, x_{k+1}^{r'} | \mathcal{H}_k, z^{VO}) \propto \mathbb{P}(x_k^{r'} | \mathcal{H}_k) \mathbb{P}(z^{VO} | x_k^{r'}, x_{k+1}^{r'}), \quad (26)$$

where $\mathbb{P}(x_k^{r'} | \mathcal{H}_k) = \mathcal{N}(\cdot, \Sigma_k)$ describes the posterior over $x_k^{r'}$ at planning time k , which could be obtained e.g. via $\mathbb{P}(x_k^{r'} | \mathcal{H}_k) = \int_{x_k^{r'}} \mathbb{P}(x_k^{r'}, x_{k+1}^{r'} | \mathcal{H}_k)$.

We now show the posterior over $x_k^{r'}$ is not influenced by the new information (measurement z^{VO}), e.g. the covariance does not change. Performing standard maximum a posteriori (MAP) inference yields the following least-squares expression:

$$x_k^{r'*}, x_{k+1}^{r'*} = \arg \min_{x_k^{r'}, x_{k+1}^{r'}} \|x_k^{r'} - \hat{x}_k^{r'}\|_{\Sigma_k}^2 + \|z^{VO} - h^{VO}(x_k^{r'}, x_{k+1}^{r'})\|_{\Sigma_{VO}}^2, \quad (27)$$

where h^{VO} and Σ_{VO} are the corresponding measurement function and measurement noise covariance for visual odometry (see e.g. Indelman et al. 2013). Linearizing and augmenting the Jacobians we get

$$\Delta x_k^{r'*}, \Delta x_{k+1}^{r'*} = \arg \min_{\Delta x_k^{r'}, \Delta x_{k+1}^{r'}} \|A \begin{pmatrix} \Delta x_k^{r'} \\ \Delta x_{k+1}^{r'} \end{pmatrix} - b\|^2, \quad (28)$$

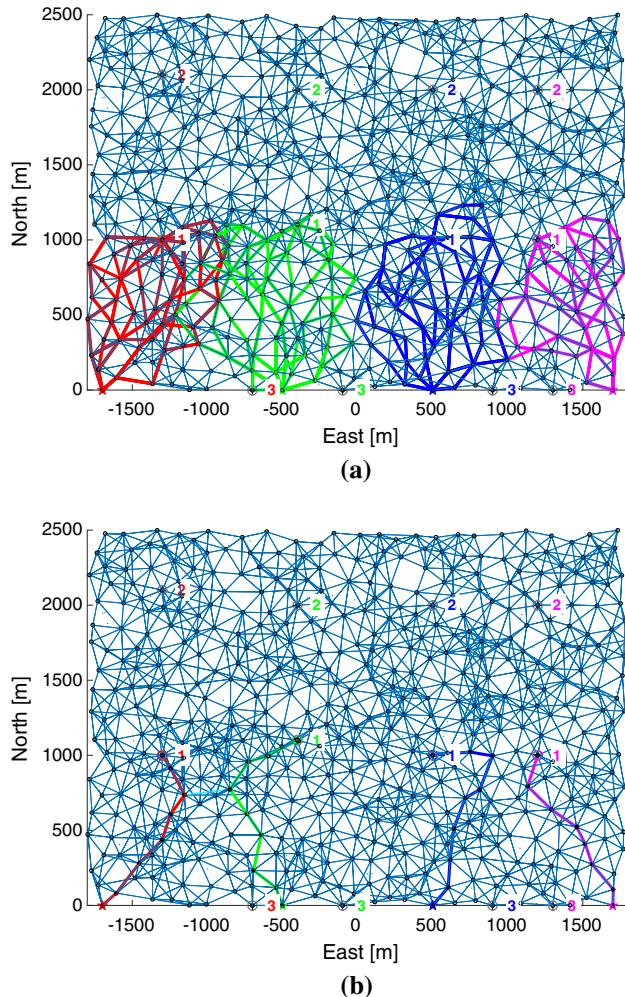


Fig. 8 Scenario1. First planning session. States of different robots are not correlated. **a** Candidate paths to the first goal of each robot; **b** Chosen paths by the planning approach

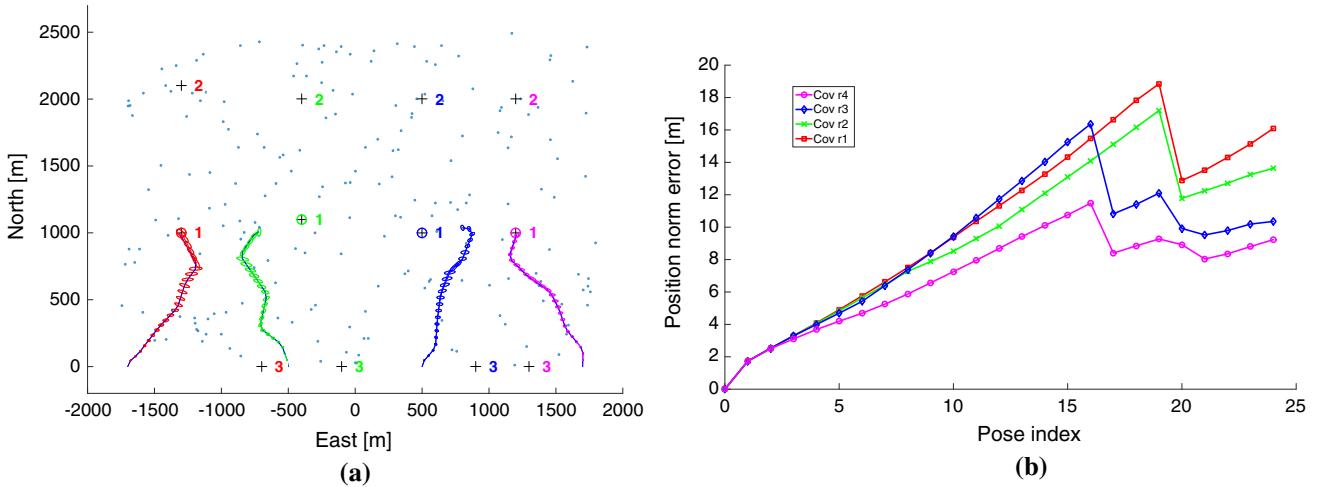


Fig. 9 Scenario 1. **a** Multi-robot SLAM given paths determined in the first planning session. The *tiny dots* represent a simplified environment in terms of landmarks, some of which are being observed during

where b is an appropriate right hand side (rhs) vector and

$$A \doteq \begin{bmatrix} \Sigma_k^{-1/2} & 0 \\ \Sigma_{VO}^{-1/2} H_{VO} - \Sigma_{VO}^{-1/2} & \end{bmatrix}. \quad (29)$$

The posterior covariance over $x_k^{r'}$ and $x_{k+1}^{r'}$ is $(A^T A)^{-1}$, from which we will now extract the entry that corresponds to $x_k^{r'}$ and show it is equal to Σ_k , despite the new information (measurement z^{VO}).

First, the information matrix is calculated as

$$A^T A = \begin{bmatrix} \Sigma_k^{-1} + H_{VO}^T \Sigma_{VO}^{-1} H_{VO} & H_{VO}^T \Sigma_{VO}^{-1} \\ \Sigma_{VO}^{-1} F & \Sigma_{VO}^{-1} \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix}. \quad (30)$$

The covariance entry that corresponds to $x_k^{r'}$ is the top left block matrix of $(A^T A)^{-1}$. Using block matrix inversion this entry can be calculated as

$$(A^T A)^{-1} = \begin{bmatrix} (\mathcal{A} - \mathcal{B}\mathcal{D}^{-1}\mathcal{C})^{-1} & \times \\ \times & \times \end{bmatrix} \quad (31)$$

Substituting matrices \mathcal{A} , \mathcal{B} , \mathcal{C} and \mathcal{D} from Eq. (30) and performing basic algebraic manipulation we get

$$\begin{aligned} (\mathcal{A} - \mathcal{B}\mathcal{D}^{-1}\mathcal{C})^{-1} &= (\Sigma_k^{-1} + F^T \Sigma_w^{-1} F)^{-1} \\ - F^T \Sigma_w^{-1} \Sigma_w^{-1} \Sigma_w^{-1} F \end{aligned} = \Sigma_k, \quad (32)$$

as claimed. In other words, adding new relative information does not impact the state $x_k^{r'}$. Hence, it does not matter whether this new information is added due to path $\mathcal{P}^{r'}$ or $\mathcal{P}_{new}^{r'}$ - in both cases, the state x_k^r of robot r is not impacted despite the existence of prior correlation between x_k^r and $x_k^{r'}$. This means, in turn, that all candidate paths \mathcal{P}^r of robot r that do not have multi-robot factors with $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$, can remain unmarked and should not be recalculated.

This concludes the simple example; we now proceed to discuss a more general case, where the covariance over $x_k^{r'}$ does change as a result of incorporating new information along a candidate path, and we outline a slight modification of our algorithm to also handle this case.

5.2 A more general case

When the conditions mentioned in Sect. 5 are not met, e.g. at least one of the paths \mathcal{P}^r , $\mathcal{P}^{r'}$ or $\mathcal{P}_{new}^{r'}$ go through previously mapped areas, or when along $\mathcal{P}^{r'}$ or $\mathcal{P}_{new}^{r'}$ there are a priori known landmarks or available GPS signal, then Eq. (25) does not necessarily hold. Intuitively, a substantial update along $\mathcal{P}^{r'}$ (or $\mathcal{P}_{new}^{r'}$), e.g. due to GPS measurement, will impact the posterior over $x_k^{r'}$:

$$\mathbb{P}(x_k^{r'} | \mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \neq \mathbb{P}(x_k^{r'} | \mathcal{H}_k). \quad (33)$$

Due to prior correlation, that couples x_k^r with $x_k^{r'}$, the new information will also pass, to some degree, onward to robot r , impacting the posterior over x_k^r . If the information along the previous and new announced paths $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$ is substantially different, then the impact on the posterior of x_k^r can also be different, i.e.:

$$\begin{aligned} & \mathbb{P}(x_k^r | \mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \\ = & \int_{x_k^{r'}} \mathbb{P}(x_k^r, x_k^{r'} | \mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \end{aligned} \quad (34)$$

$$\doteq \mathcal{N}(\times, \Sigma_k(U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'}))) \quad (35)$$

$$\begin{aligned} & \mathbb{P}(x_k^r | \mathcal{H}_k, U^{r'}(\mathcal{P}_{new}^{r'}), Z^{r'}(\mathcal{P}_{new}^{r'})) \\ = & \int_{x_k^{r'}} \mathbb{P}(x_k^r, x_k^{r'} | \mathcal{H}_k, U^{r'}(\mathcal{P}_{new}^{r'}), Z^{r'}(\mathcal{P}_{new}^{r'})) \end{aligned} \quad (36)$$

$$\doteq \mathcal{N}(\times, \Sigma_k(U^{r'}(\mathcal{P}_{new}^{r'}), Z^{r'}(\mathcal{P}_{new}^{r'}))) \quad (37)$$

and

$$\mathbb{P}(x_k^r | \mathcal{H}_k, U^{r'}(\mathcal{P}^{r'}), Z^{r'}(\mathcal{P}^{r'})) \neq \mathbb{P}(x_k^r | \mathcal{H}_k, U^{r'}(\mathcal{P}_{new}^{r'}), Z^{r'}(\mathcal{P}_{new}^{r'})). \quad (38)$$

Hence, the posteriors over candidate path \mathcal{P}^r , $b[\mathcal{P}^r]$ and $b'[\mathcal{P}^r]$ will change (Eq. (25) will not hold). It would thus seem that \mathcal{P}^r should be necessarily marked, to trigger belief evolution recalculation.

However, it is often the case that while the posteriors (34) and (36), and therefore $b[\mathcal{P}^r]$ and $b'[\mathcal{P}^r]$, are not identical, in practice the difference is small and can be considered negligible given some threshold. In such a case, there is no need in recalculating belief evolution along path \mathcal{P}^r , and thus the latter should *not* be marked.

Based on the above observation, we propose the following slight modification to our approach. First, we evaluate the posteriors (34) and (36) - this is a one-time calculation for given previous and new announced paths \mathcal{P}^r and $\mathcal{P}_{new}^{r'}$, which is valid to *all* candidate paths \mathcal{P}^r of robot r . Then, we decide if the two posteriors are sufficiently similar given a user-defined threshold th : different information-theoretic costs can be used for this purpose (e.g. KL-divergence and relative entropy). A simple alternative, for example, is to calculate the difference in the determinant (or trace) of the posterior covariance in each case. More specifically, recalling Eqs. (35) and (37), the candidate path \mathcal{P}^r is marked only if

$$\begin{aligned} & \det(\Sigma_k(U^{r'}(\mathcal{P}_{new}^{r'}), Z^{r'}(\mathcal{P}_{new}^{r'}))) \\ & - \det(\Sigma_k(U^{r'}(\mathcal{P}_{new}^{r'}), Z^{r'}(\mathcal{P}_{new}^{r'}))) > th. \end{aligned} \quad (39)$$

In our current implementation we use the above criteria with the threshold th set to 10. We note that retrieving the marginal covariances involved in Eq. (39) can be done efficiently from the (square root) information matrix (Golub and Plemmens 1980). Another alternative might be to utilize our recently developed variant of the matrix determinant lemma (Kopitkov and Indelman 2016, 2017) to make these one-time calculations even more efficient.

6 Computational complexity aspects

Exhaustively evaluating all candidate path combinations between different robots results in exponential computational complexity in number of robots. Resorting to announced path approach (Levine et al. 2013; Atanasov et al. 2015) reduces this complexity to linear, at the cost of sub-optimality, i.e. $O(R \cdot M \cdot T) \cdot O(B_{full})$ where R is the number of robots, M is the number of candidate paths of each robot, and T denotes the number of iterations until convergence. $O(B_{full})$ represents the computational complexity of evaluating the objective function (5) given a candidate path and announced path from other robots. In particular, $O(B_{full})$ involves inference over the joint belief for appropriate paths, and retrieval of a covariance term as required by the objective

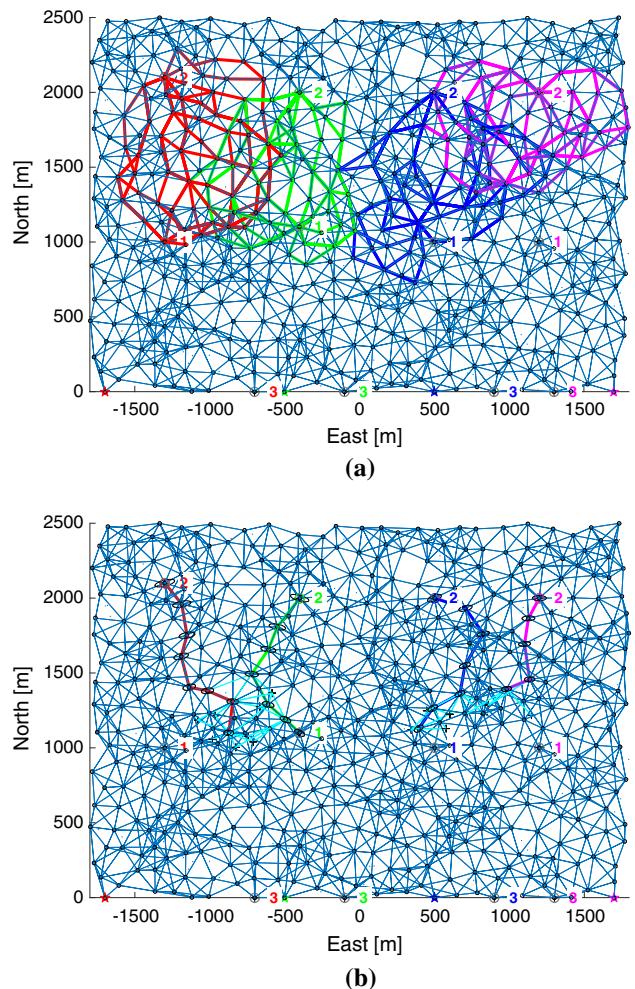


Fig. 10 Scenario 01. Second planning session. **a** Candidate paths to the second goal of each robot; **b** Chosen paths by the planning approach, including uncertainty propagation along each path that corresponds to multi-robot beliefs from appropriate look ahead steps. The figure also shows cyan observations of previously-mapped landmarks and multi-robot indirect constraints. (Color figure online)

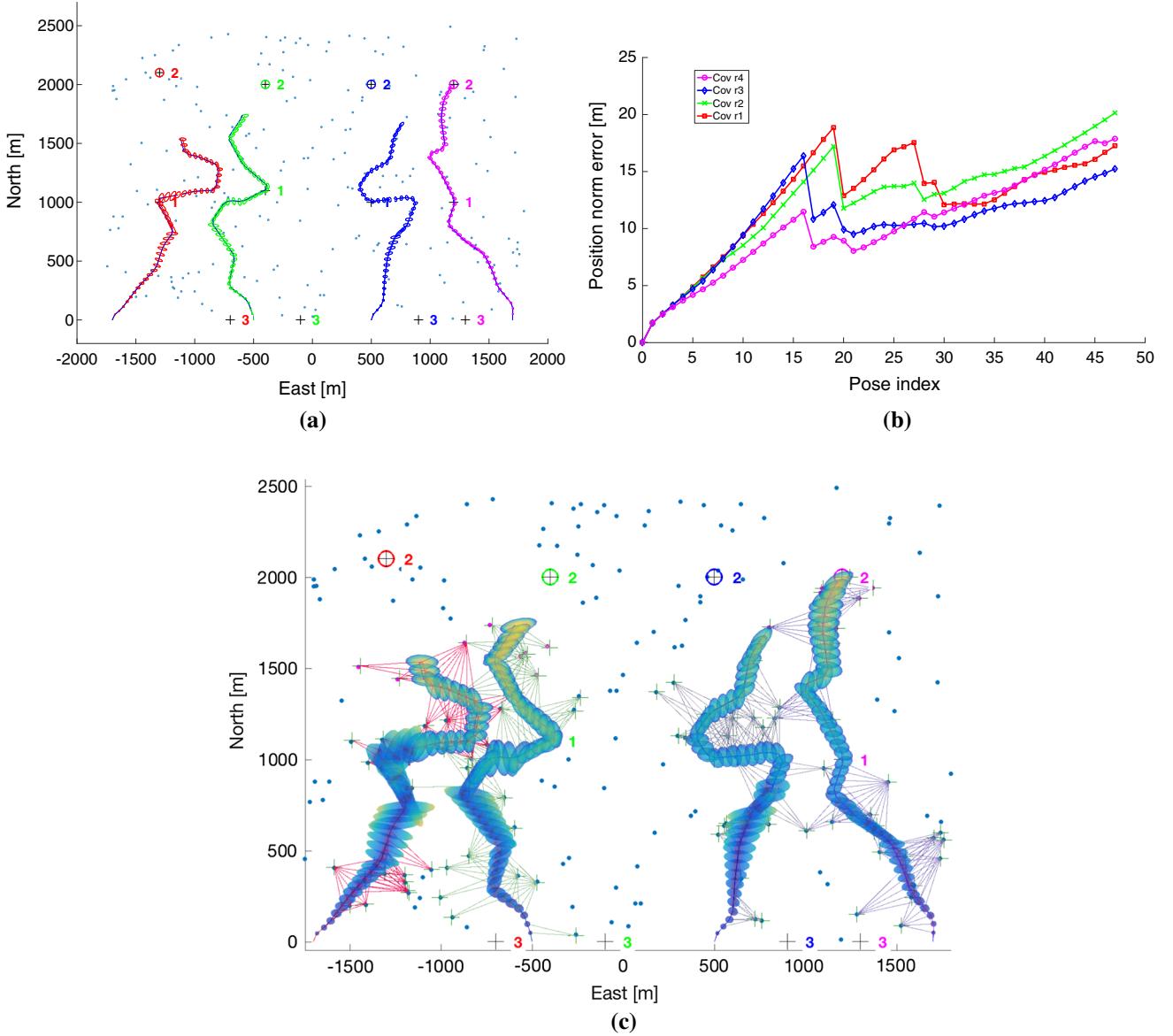


Fig. 11 Scenario 1. **a** Multi-robot SLAM given paths determined in the second planning session. The *tiny dots* represent a simplified environment in terms of landmarks, some of which are being observed during SLAM as shown in **(c)**. **b** Corresponding position covariance evolution as a function of time (represented discretely by pose index in

simulation). **c** SLAM visualization of **(a)**, including explicit representation of landmark observations. Uncertainty covariances were inflated for visualization purposes. Note the mutual landmark observations by different robots

function (5). While previous announced approaches, such as Atanasov et al. (2015), perform calculations efficiently by exploiting sparsity, no calculation re-use is done between different candidate actions, and therefore $O(B_{full})$ is the complexity for evaluating each action.

In contrast, the proposed approach reduces $O(B_{full})$, starting from the second iteration, by identifying non-impacted paths and re-using calculations for impacted paths. We shall denote this reduced complexity by $O(B)$.

Specifically, calculating the objective function for non-impacted paths involves a one-time calculation of $\Delta J'$ that

only depends on the length, in terms of number of nodes in PRM, of the previous and new announced paths (see Sect. 4.1, and line 11 in Algorithm 2). We thus denote this complexity as $O(L(\mathcal{P}))$. Given this one time calculation, the per-action complexity is $O(B) = O(1)$.

For the impacted paths, $O(B)$ is a function of how many factors the previous and new announced paths have in common, see Eq. (21). While one could reduce this complexity even further by resorting to incremental inference (e.g. ISAM Kaess et al. 2012), in this work we consider batch inference. Under this setting, $O(B) \leq O(B_{full})$. In other words, the

computational complexity of performing (batch) inference while reusing calculations, and then retrieving the required covariances is less or equal to the same operation without re-using calculations, i.e. using Eq. (12).

To summarize, for m and $M - m$ non-impacted and impacted paths, respectively, the per-iteration computational complexity for each robot, starting from the second iteration, is reduced as follows:

$$O(MB_{full}) \rightarrow O(L(\mathcal{P})) + mO(1) + (1 - m)O(B), \quad (40)$$

with $m \in [0, M]$.

7 Experiments

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 (Dellaert 2012) to investigate key aspects of the proposed approach. The objective function (5) is

$$J = \sum_{r=1}^R \left[\kappa_{goal}^r t_{goal}^r + \kappa_{\Sigma}^r \text{tr} \left(\Sigma_{goal}^r \right) \right], \quad (41)$$

where Σ_{goal}^r and t_{goal}^r represent, respectively, the covariance upon reaching the goal and time of travel (or path length) for robot r . The parameters κ_{goal}^r and κ_{Σ}^r weight the importance of each term (we use $\kappa_{path}^r = 0.1$ and $\kappa_{uncert}^r = 1$). A probabilistic roadmap (PRM) (Kavraki et al. 1996) is used, to discretize the (partially unknown) environment and generate candidate paths over the roadmap. Unknown areas in the environment are treated in the current implementation similarly, i.e. addressed as unoccupied. Although outside the scope of this work, one can also update the roadmap and the candidate paths once new information becomes available (e.g. obstacle detection), such that only valid, non-colliding paths remain (see e.g. our previous research Indelman 2017).

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. Levine et al. 2013; Indelman 2015a; Atanasov et al. 2015). 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. In all the scenarios reported in the sequel, typically 3–4 iterations were required until convergence.

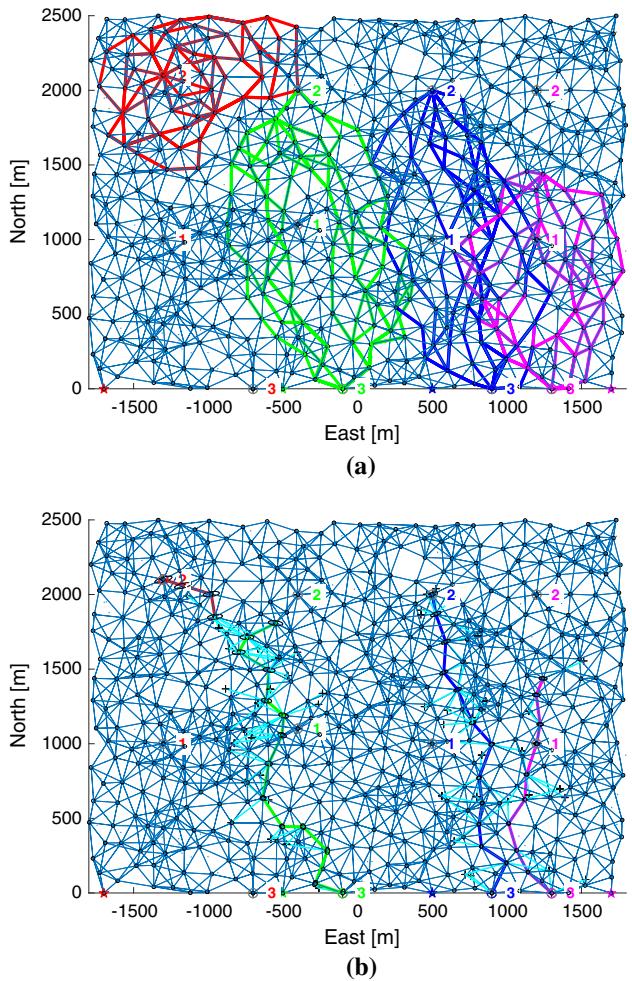


Fig. 12 Scenario 01. Third planning session. **a** Candidate paths to the second goal for the red robot, and to the third goal of each other robot; **b** Chosen paths by the planning approach, including uncertainty propagation along each path that corresponds to multi-robot beliefs from appropriate look ahead steps. The figure also shows in cyan observations of previously-mapped landmarks and multi-robot indirect constraints (Color figure online)

7.1 Basic scenario

Figure 5 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.

Figure 6b 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 Fig. 1. The corresponding belief evolution (covariance ellipses) is displayed in black. Robot r determines its best path, and announces it to other robots, which do the same; the process is repeated until convergence. Similar to Indelman (2015a), we use a simple heuristic for the function $\text{cr}_{\text{MR}}(v_i, v_j)$ (line 4 of Algorithm 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 $d = 300$ m. More advanced methods could be implemented, e.g. considering also statistical knowledge.

The set of involved vertices in PRM, V_{inv} , depicted conceptually in Fig. 2, is shown for robot r in Fig. 7 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 $\mathcal{P}_{new}^{r'}$ 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 $\mathcal{P}_{new}^{r'}$), and yellow and magenta indicate multi-robot factors that are associated, respectively, only with $\mathcal{P}^{r'}$ and $\mathcal{P}_{new}^{r'}$. These factors are appropriately then included within the corresponding vertices in V_{inv} and are used for calculating belief evolution, following Algorithms 1 and 2.

In the specific situation shown in Fig. 7, 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.

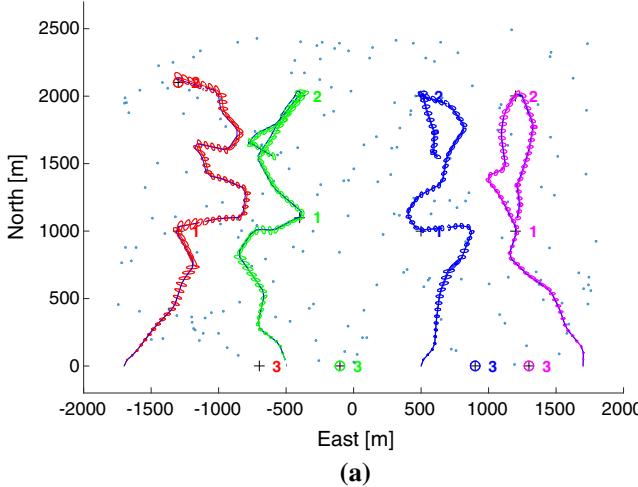
Figure 6a reports statistical timing results as a function of number of candidate paths N_{cand} for each robot, considering the two-robot and four-robot scenarios from Fig. 5. These results were obtained by running each approach 50 times, for each considered N_{cand} . In each such run, the scenario

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 were obtained, compared with the Standard approach.

7.2 Larger Scenarios

We also examine our approach in a larger scenario, Scenario1, where the world is represented by 200 (a priori unknown) landmarks, randomly scattered in a rectangular area of 2000×2500 meters squared. In this scenario each robot has to reach multiple pre-defined goals while operating in unknown environments. Such a scenario involves multiple planning sessions and multi-robot SLAM - reaching each goal triggers a new planning session during which the robots update theirs best paths. These paths are then translated into commands, in our case, the change in heading angle. In our simulative framework, the robots execute these commands and acquire new bearing and range observations of landmarks. Note that the latter can be either previously seen landmarks, that correspond to already mapped areas, and new landmarks. Considering perfect association of the landmark observations, the robots then calculate a multi-robot SLAM solution, i.e. the term $\mathbb{P}(X_k | \mathcal{H}_k)$ in Eq. (11).

Two variations (Scenario2 and Scenario3) of this scenario were also considered. These scenarios differ in the initial uncertainty (prior) and whether multi-robot factors are integrated within planning, as in the proposed approach,



(a)

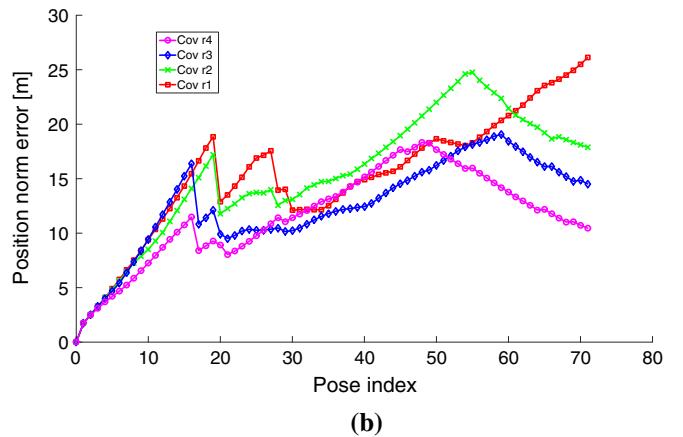


Fig. 13 Scenario1. **a** Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during

SLAM. **b** Corresponding position covariance evolution as a function of time (represented discretely by pose index in simulation)

or not. Table 1 summarizes all the three considered large-scale scenarios. We now proceed to in-depth performance evaluation of Scenario1, and then in the sequel also briefly summarize performance aspects for Scenario2 and Scenario3.

Figures 8–15 show the results of each of the planning and SLAM sessions, while the running time is reported in Fig. 16. Goals are indicated in these figures using both numbers and colors, with the former denoting sequence (i.e. goal 1 should be visited before goal 2), and colors indicating different robots. As seen, four robots are considered (red, green, blue and purple), and each robot has a sequence of three goals. We intentionally scattered the goals in such a way that both planning in unknown and previously-mapped environments is examined.

We show, for each planning session, the candidate paths for all robots and the best paths identified by the proposed approach, see Figs. 8, 10, 12 and 14. Multi-robot factors involving future poses of different robots (along the chosen paths), and factors involving a future pose of robot $r \in [1, \dots, R]$ and a landmark, previously observed by robot r or by any other robot in the group, are indicated in cyan color. See, e.g. Fig. 10b for combination of both of these factors. As in the basic study (Sect. 7.1), covariances along the chosen paths are also shown.

At the first planning session (Fig. 8), the robot start operating with only prior information on their initial poses (we use $1e-6 [m]$, meaning robots know their exact start locations) - in other words, there is no correlation between the robot states. Using the proposed approach, the best path for each robot in the group is determined and executed until one of the robots reaches a goal. In particular, the chosen paths of the red and green robots admit a single multi-robot factor within planning. Figure 9a shows the corresponding SLAM solution, while Fig. 11c shows position covariance evolution (from SLAM). While not explicitly shown, the states of red and green robots, and of blue and purple robots become correlated towards the end of this phase due to mutual landmark observations.

From this moment onward, thus, the states of these robots are (somewhat) correlated and the discussion from Sect. 5 regarding prior correlation becomes relevant. In the second planning session (Figs. 10 and 11), the goals are scattered such that vast majority of the candidate paths still go through unknown areas (see Fig. 10a). Looking at the determined best paths (Fig. 10b), one can observe the planned multi-robot collaboration between two robot pairs (red-green and blue-purple), which is exhibited either in terms of multi-robot factors or observations of landmarks previously observed by another robot.

Despite prior correlation, however, our approach is capable of significantly reducing running time (by a factor of two, see Fig. 16) while yielding the same results in terms of the

chosen paths. This goes in hand with the observation from Sect. 5 that the belief along path of any robot r is not impacted by change in the announced path of other robots if these paths go through unknown areas and without sources of absolute information, which is the case here (recall also the example from Sect. 5.1).

In the third planning session, the red robot still has not reached its second goal, while all the other robots already consider their next goals. After the red robot reaches its second goal, another planning session is triggered. We note that in practice, only the red robot could actually generate new candidate paths while the rest of the robots could remain with candidate paths from the previous planning session.

The third goal of each robot was intentionally chosen to force the robots to re-visit previously mapped environments (see e.g. Figs. 14 and 15). As in the previous planning

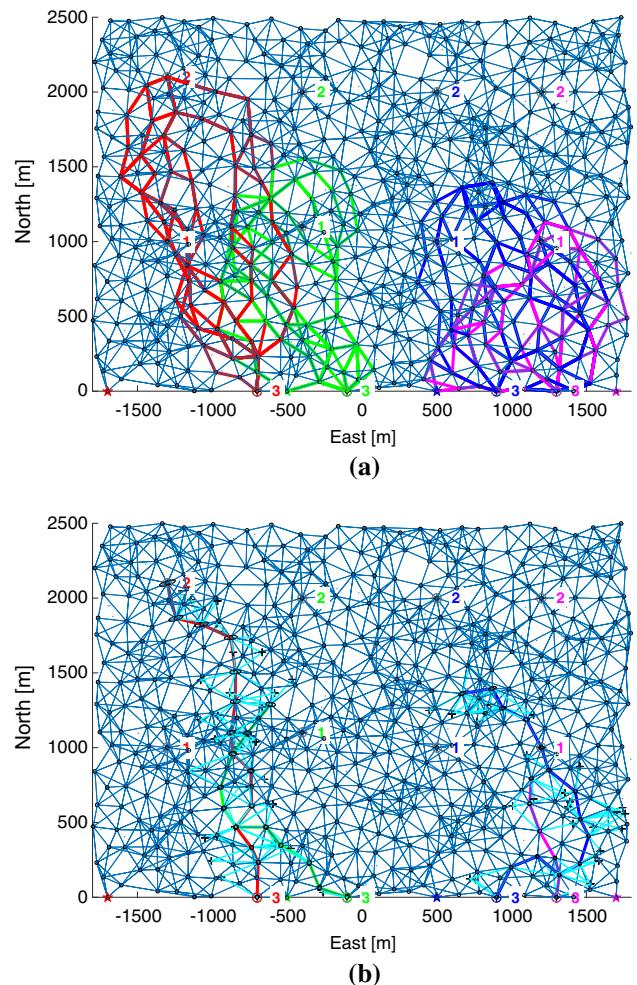


Fig. 14 Scenario1. Fourth planning session. **a** Candidate paths to the third goal of each robot; **b** Chosen paths by the planning approach , including uncertainty propagation along each path that corresponds to multi-robot beliefs from appropriate look ahead steps. The figure also shows in cyan observations of previously-mapped landmarks and multi-robot indirect constraints (Color figure online)

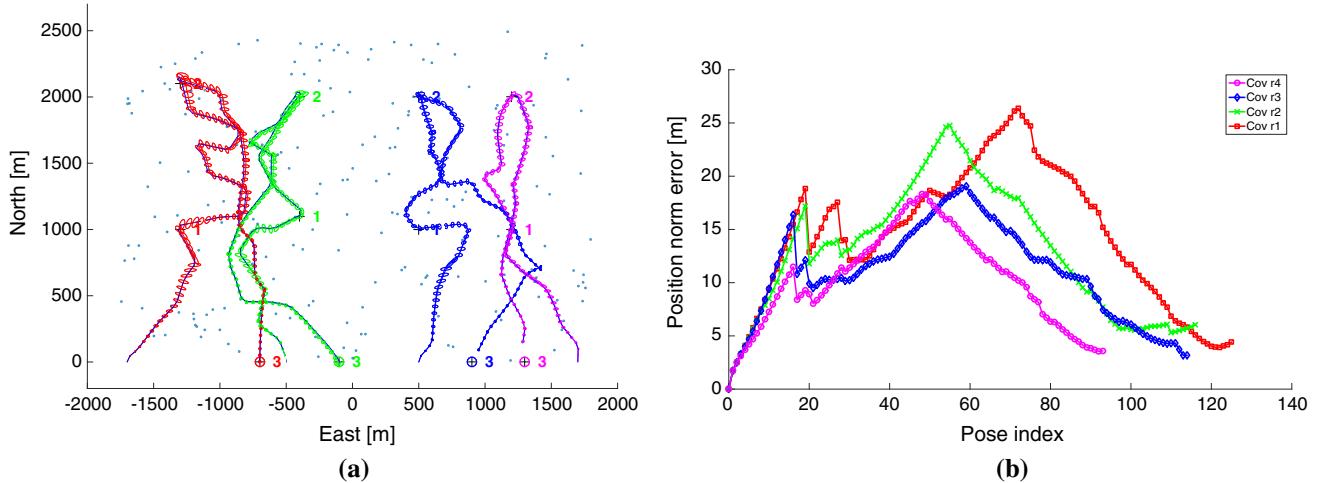


Fig. 15 Scenario 1. **a** Multi-robot SLAM given paths determined in the fourth planning session. The *tiny dots* represent a simplified environment in terms of landmarks, some of which are being observed during

SLAM. **b** Corresponding position covariance evolution as a function of time (represented discretely by pose index in simulation)

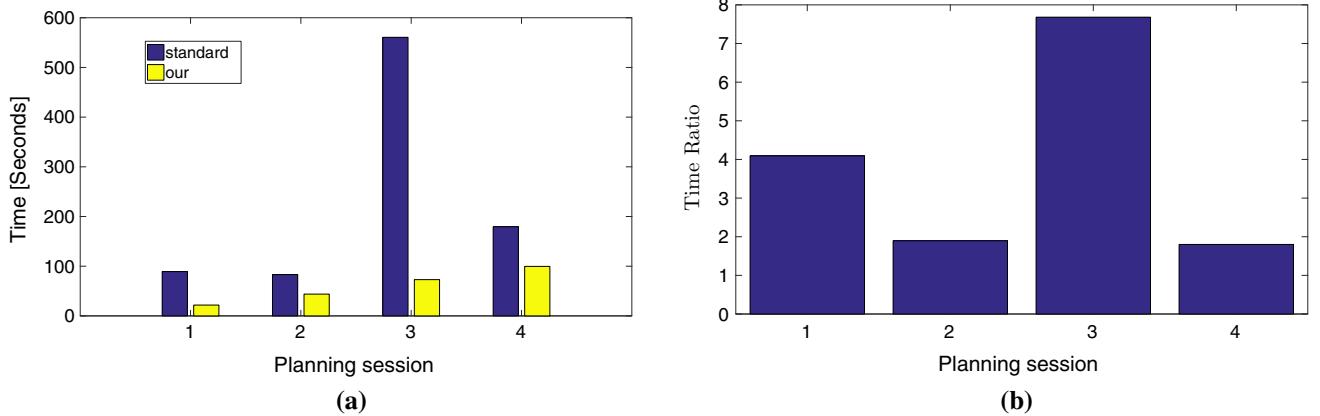


Fig. 16 Scenario 1. Running time comparison between the proposed and the Standard approach. **a** Running time for each planning session. **b** Ratio of running time of the proposed and standard approaches

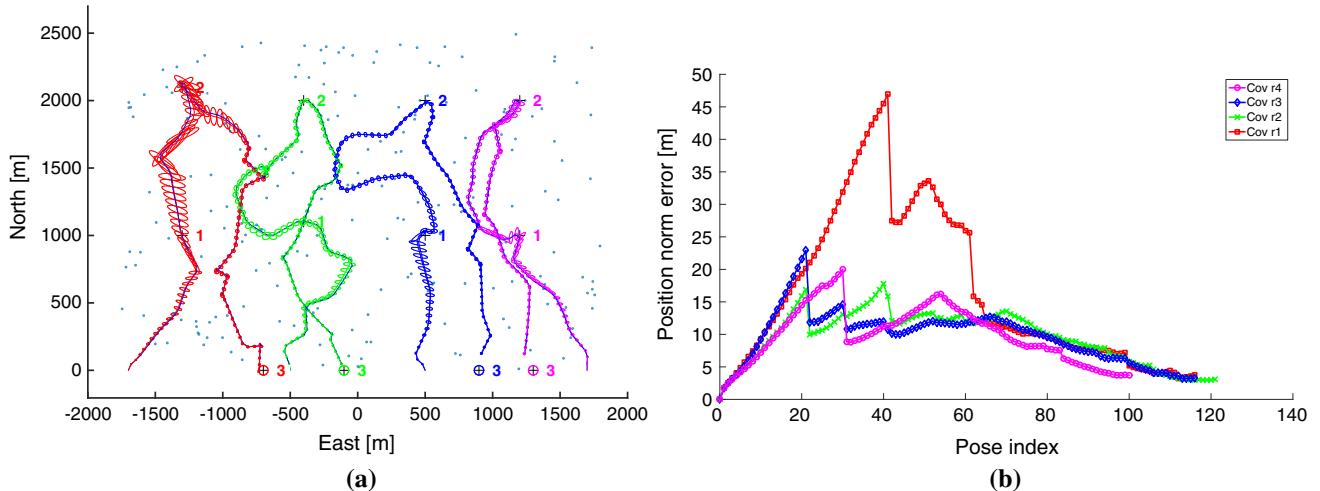


Fig. 17 Scenario 2. **a** Multi-robot SLAM given paths determined in the third planning session. The *tiny dots* represent a simplified environment in terms of landmarks, some of which are being observed during SLAM. **b** Corresponding position covariance evolution

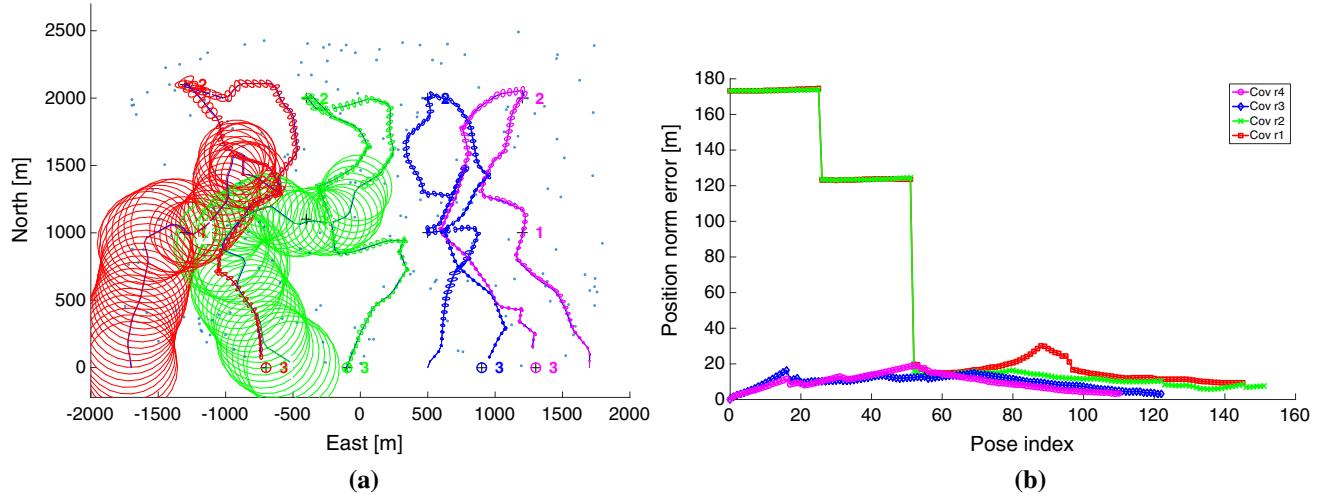


Fig. 18 Scenario3. **a** Multi-robot SLAM given paths determined in the third planning session. The tiny dots represent a simplified environment in terms of landmarks, some of which are being observed during

SLAM. **b** Corresponding position covariance evolution as a function of time (represented discretely by pose index in simulation)

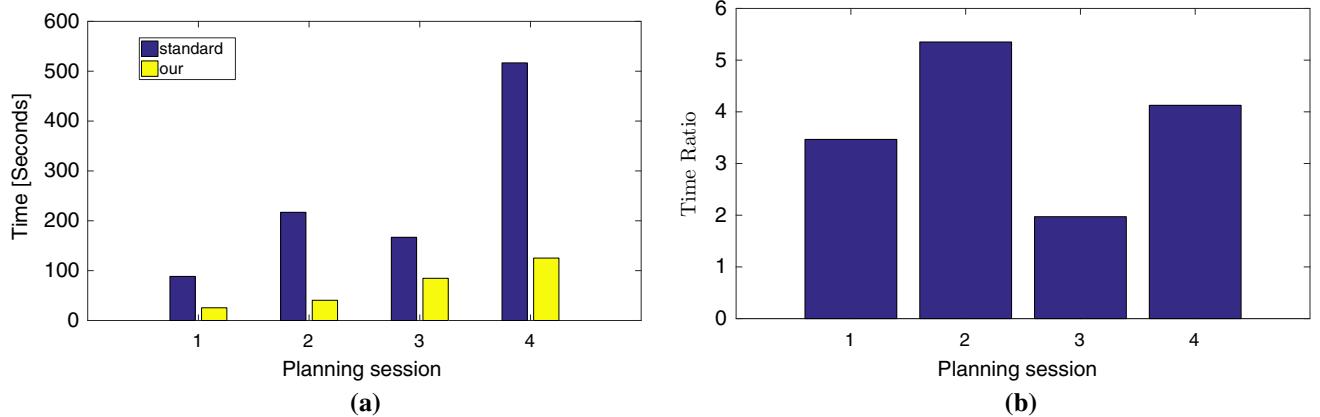


Fig. 19 Scenario3. Running time comparison between the proposed and the Standard approach. **a** Running time for each planning session. **b** Ratio of running time of the proposed and standard approaches

sessions, the states of the two robot pairs red-green and blue-purple are correlated. However, here, in addition the robots consider impact of loop closure observations within planning. These are often mutual multi-robot observations, i.e. the same previously-observed landmarks are planned to be observed by multiple robots - see the cyan lines in Fig. 14b. Given the corresponding best paths, which were determined as such (mainly) due to these multi-robot constraints that allow significant uncertainty reduction, a multi-robot SLAM session is performed. As evident from Fig. 15, the robots indeed reach the goals with small uncertainty, which roughly corresponds to the prior uncertainty (due to loop closures).

Finally, Fig. 16 depicts running time for each of the planning sessions, comparing the proposed approach with the Standard approach (that does not attempt to re-use calcula-

tions). It can be clearly seen that our approach is substantially faster in all planning sessions. In particular, it is faster by a factor of two and seven in the second and third planning sessions, respectively. We emphasize this significant reduction in running time comes with no sacrifice in performance, i.e. the same paths were chosen by our and Standard approach in all planning sessions.

We now summarize main aspects also for the other two large scale scenarios, Scenario2 and Scenario3 (see Table 1).

In Scenario2, all robots have (small) identical prior covariances, as in Scenario1. However, in this scenario the robots do *not* use MR factors within planning (multi-robot SLAM is still performed). The obtained multi-robot SLAM results given the calculated paths by this variation

of the planning approach are shown in Fig. 17a, while Fig. 17b depicts the corresponding covariance evolution. Comparing the covariance in Fig. 15b, which was obtained with MR factors within planning (the proposed approach), one can clearly observe uncertainty reaches significantly higher values in Fig. 17b (without using MR factors within planning).

In Scenario3, red and green robots have larger initial uncertainty covariances, while the other two robots (blue and purple) have a small initial covariance as in Scenario1. In both cases, all robots use MR factors within planning. Figure 18 shows the results and the covariance evolution, while Fig. 19 compares running time between the proposed approach and the Standard approach. As seen, the large uncertainties of the red and green robots are reduced due to mutual observations with the blue robot, observations that were planned by the proposed approach. As earlier, running time of the latter is significantly smaller than of the Standard approach, while in both cases the same results are obtained.

8 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. Our approach is applicable when states of different robots are statistically independent at planning time, but also in presence of correlation, e.g. due to previous mutual landmark observations, which is expected in practical real world scenarios. 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. Specifically, in large scale simulated scenarios computation time of planning phase was reduced by a factor ranging between $\times 2$ and $\times 8$.

Future work will focus on performance evaluation in real-world experiments as well as improving different aspects of the proposed method. The latter include resorting to incremental factorization approaches to make evaluation of

impacted paths even more computationally efficient, reducing further run-time by avoiding explicitly evaluating the poster joint belief (Kopitkov and Indelman 2016, 2017), and re-using calculations between (multi-robot) inference and planning phases (Farhi and Indelman 2017).

While the method currently uses PRM to generate the candidate actions, any-time planning algorithms (e.g. RRT) may be more appropriate for online operation in real world settings. We believe the computational advantages of the proposed method would apply directly in that framework as well, and leave further investigation of these aspects to future research.

Acknowledgements 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.

References

- Agha-Mohammadi, A.-A., Chakravorty, S., & Amato, N. M. (2014). Firm: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *The International Journal of Robotics Research*, 33, 268–304.
- Agha-mohammadi, A. A., Agarwal, S., Chakravorty, S., & Amato, N. M. (2015). Simultaneous localization and planning for physical mobile robots via enabling dynamic replanning in belief space. *arXiv:1510.07380*.
- Agha-Mohammadi, A. A., Agarwal, S., Mahadevan, A., Chakravorty, S., Tomkins, D., Denny, J., et al. (2014). Robust online belief space planning in changing environments: Application to physical mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 149–156).
- Amato, C., Konidaris, G., Anders, A., Cruz, G., How, J. P., & Kaelbling, L. P. (2016). Policy search for multi-robot coordination under uncertainty. *The International Journal of Robotics Research*, 35(14), 1760–1778.
- Atanasov, N., Le Ny, J., Daniilidis, K., Pappas, G. J. (2015). Decentralized active information acquisition: Theory and application to multi-robot slam. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- Bernstein, D., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4), 819–840.
- Bry, A., Roy, N. (2011). Rapidly-exploring random belief trees for motion planning under uncertainty. In *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 723–730).
- Chaves, S. M., Kim, A., Eustice, R. M. (2014). Opportunistic sampling-based planning for active visual slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3073–3080). IEEE.
- Cunningham, A., Indelman, V., Dellaert, F. (2013). DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany.
- Dellaert, F. (2012). Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology.
- Farhi, E. I., Indelman, V. (2017) Towards efficient inference update through planning via jip - joint inference and belief space planning.

- In *IEEE International Conference on Robotics and Automation (ICRA)*.
- Golub, G. H., & Plemmons, R. J. (1980). Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition. *Linear Algebra and Its Applications*, 34, 3–28.
- Hollinger, G. A., & Sukhatme, G. S. (2014). Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33, 1271–1287.
- Indelman, V. (2015a). Towards cooperative multi-robot belief space planning in unknown environments. In *Proceedings of the International Symposium of Robotics Research (ISRR)*.
- Indelman, V. (2015b). Towards multi-robot active collaborative state estimation via belief space planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Indelman, V., Carlone, L., & Dellaert, F. (2015). Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research*, 34(7), 849–882.
- Indelman, V., Gurfil, P., Rivlin, E., & Rotstein, H. (2012). Graph-based distributed cooperative navigation for a general multi-robot measurement model. *The International Journal of Robotics Research*, 31(9), 1057–1080.
- Indelman, V., Williams, S., Kaess, M., & Dellaert, F. (2013). Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61(8), 721–738.
- Indelman, V. (2017). Cooperative multi-robot belief space planning for autonomous navigation in unknown environments. *Autonomous Robots*. doi:[10.1007/s10514-017-9620-6](https://doi.org/10.1007/s10514-017-9620-6).
- Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31, 217–236.
- Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7), 846–894.
- Kavraki, L. E., Svestka, P., Latombe, J.-C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 566–580.
- Kopitkov, D., Indelman, V. (2016). Computationally efficient decision making under uncertainty in high-dimensional state spaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Kopitkov, D., & Indelman, V. (2017). Computationally efficient belief space planning via augmented matrix determinant lemma and reuse of calculations. *IEEE Robotics and Automation Letters (RA-L)*, 2(2), 506–513.
- Kurniawati, H., Hsu, D., Lee, W. S. (2008). Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems (RSS)* (vol. 2008).
- LaValle, S. M., & Kuffner, J. J. (2001). Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5), 378–400.
- Levine, D., Luders, B., & How, J. P. (2013). Information-theoretic motion planning for constrained sensor networks. *Journal of Aerospace Information Systems*, 10(10), 476–496.
- Olson, E., & Agarwal, P. (2013). Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, 32(7), 826–840.
- Papadimitriou, C., & Tsitsiklis, J. (1987). The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3), 441–450.
- Pathak, S., Thomas, A., Feniger, A., & Indelman, V. (2016). Robust active perception via data-association aware belief space planning. [arXiv:1606.05124](https://arxiv.org/abs/1606.05124).
- Platt, R., Tedrake, R., Kaelbling, L.P., & Lozano-Pérez, T. (2010). Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems (RSS)* (pp. 587–593). Zaragoza, Spain.
- Prentice, S., & Roy, N. (2009). The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, 28, 1448–1465.
- Regev, T., & Indelman, V. (2016). Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Stachniss, C., Grisetti, G., & Burgard, W. (2005). Information gain-based exploration using rao-blackwellized particle filters. In *Robotics: Science and Systems (RSS)* (pp. 65–72).
- Van Den Berg, J., Patil, S., & Alterovitz, R. (2012). Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11), 1263–1278.



Tal Regev is pursuing an M.Sc. degree in Computer Science Department at the Technion - Israel Institute of Technology. He holds a B.Sc. degree in Computer Science, awarded by the Technion in 2013. Tal's current research interests include autonomous multi-robot navigation, mapping and perception.



Vadim Indelman is currently an Assistant Professor in the Department of Aerospace Engineering at the Technion, and he is also a member of the Technion Autonomous Systems Program (TASP). Upon joining the Technion in July 2014 Dr. Indelman established the Autonomous Navigation and Perception Laboratory (ANPL), where he and his research group investigate problems related to single and multi-robot collaborative autonomous navigation and perception, with a particular focus on online, accurate and reliable operation in uncertain and unknown environments. Prior to joining the Technion as a faculty member, Dr. Indelman was a postdoctoral fellow in the Institute of Robotics and Intelligent Machines (IRIM) at the Georgia Institute of Technology (between 2012 and 2014). He obtained his Ph.D. degree in Aerospace Engineering at the Technion - Israel Institute of Technology in 2011, and also holds B.A. and B.Sc. degrees in Computer Science and Aerospace Engineering, respectively, both awarded by the Technion in 2002.