Consistent Sparsification for Efficient Decision Making Under Uncertainty in High Dimensional State Spaces

Khen Elimelech and Vadim Indelman

***Abstract*— In this paper we introduce a novel approach for efficient decision making under uncertainty and belief space planning, in high dimensional state spaces. While recently developed methods focus on sparsifying the inference process, the sparsification here is done in the context of efficient decision making, with no impact on the state inference. By identifying state variables which are uninvolved in the decision, we generate a sparse version of the state's information matrix, to be used in the examination of candidate actions. This sparse approximation is action-consistent, i.e. has no influence on the action selection. Overall we manage to maintain the same quality of solution, while reducing the computational complexity of the problem. The approach is put to the test in a SLAM simulation, where a significant improvement in runtime is achieved. Nevertheless, the method is generic, and not tied to a specific type of problem.**

I. Introduction

In recent years, modern robots have been breaking the barriers of laboratories into the real world. In real scenarios, accounting for different sources of uncertainty is essential, both in state inference and decision making, such that truly autonomous, reliable and robust performance can be attained. Moreover, such settings often involve reasoning about a belief over a high-dimensional state. Relevant problems include SLAM and autonomous navigation in unknown environments. Causes for the uncertainty in that context can be, for instance, noisy measurements, failed locomotion attempts (e.g. wheels slippage) and inherit risk taking.

Decision making under uncertainty has been investigated and demonstrated in a broad range of contexts, including sensor deployment (e.g. [9]), active sensing (e.g. [16]) and autonomous navigation and active SLAM (e.g. [8], [5], [11]), as stated. It was also found applicable to high level problems such as natural language interpretation and financial real-time decision making agents.

The uncertainty is expressed using ideas from the world of information theory. Terminology-wise, we can talk about belief space planning (BSP), where the belief represents a probability density function over the state (e.g. robot poses, mapped environment) given the actions and observations obtained thus far. Numerous BSP approaches have been recently developed in an effort to reduce computational complexity at the cost of sub-optimal performance, see e.g. [1], [13]–[15], [17], [18], [20].

K. Elimelech is with the Robotics and Autonomous Systems Program (TASP), Technion - Israel Institute of Technology, Haifa 32000, Israel, khen@campus.technion.ac.il. V. Indelman is with the Department of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel, vadim.indelman@technion.ac.il. This work was supported by the Israel Science Foundation.

BSP and decision making in high dimensional states space is in particular computationally expensive. Decision making for a belief over n dimensional state, involves evaluating an information-theoretic metric, such as entropy, for numerous candidate actions. Each of these requires calculating a determinant of $n \times n$-sized posterior information (or covariance) matrix. Overall, the amount of information which has to be processed for every decision can be enormous. As a result, *online* decision making in high-dimensional state spaces is challenging, especially considering computationally-constrained robots, such as quadrotors.

Due to its importance in modern robotics, the topic has gained more interest in recent years, in attempt to deal with the high computational cost. However, state of the art approaches typically focus on the passive instance of the problem, i.e. inference. In context of SLAM, numerous approaches exploit sparsity of the underlying information space, while re-using calculations when possible (see e.g. [6], [10]). Moreover, sparsification approaches have been recently developed to extend online performance capabilities in the context of long-term autonomy [2], [3], [12], [19]. These approaches typically solve a convex optimization problem using KL distance as the metric. Yet, the main focus of the above sparsification approaches was for the inference process and not for the decision making stage.

A recent publication has investigated a sparsification method at the level of decision making [4]. According to that approach, correlations between variables can sometimes be ignored, when comparing candidate actions, without sacrificing performance. Yet that approach is limited to greedy decision making with unary observation models, limitations which are often hard to cope with in a realistic scenario. Still, we consider that approach as a preliminary proof of concept, which could trigger further research in this unexplored area.

In this paper we present the first well founded attempt to optimize the decision making process directly, while considering a general problem setting comprising arbitrary transition and observation models, and possibly non-myopic setup. Our approach significantly sparsifies the underlying information matrices (see Fig. 1) in such a way that does not impact the decision making process, while substantially reducing runtime. In other words, despite the sparsification, results are *action consistent*, as to be explained ahead.

As a part of the approach, we also introduce a solid standardization to several ideas in decision making and its optimization, previously undefined. This system of definitions allows to clearly compare the quality of different approaches, and opens the door to many possible future directions.

II. Notations and Problem Formulation

Consider the posterior distribution at time k over state X_k

$$b[X_k] \doteq \mathbb{P}(X_k \mid a_{0:k-1}, z_{0:k}), \tag{1}$$

where $a_{0:k-1} \doteq \{a_0, \ldots, a_{k-1}\}$ and $z_{0:k} \doteq \{z_0, \ldots, z_k\}$ are the actions and observations until time k, respectively. The variables in the state vector X_k are problem-specific. In the context of navigation, it can be defined, for example, as $X_k \doteq \{x_0, \ldots, x_k\}$, where x_i denotes the robot pose at time i. For SLAM problems, it can also include locations of observed landmarks. To describe such a distribution we use the term *belief*.

Considering the following transition and observation models

$$x_{k+1} = g(x_k, a_k) + w_k, \quad w_k \sim \mathcal{N}(0, W_k) \tag{2}$$
$$z_k = h(X_k^o) + v_k, \quad v_k \sim \mathcal{N}(0, V_k), \tag{3}$$

both containing some *Gaussian* noise. In Eq. (3), the notation $X_k^o \subseteq X_k$ indicates the directly involved variables in the observation model. These models can be described probabilistically as $\mathbb{P}(x_{k+1} \mid x_k, a)$ and $\mathbb{P}(z^k \mid X_k^o)$.

One can now reason about the belief at a future time $k+1$ given some action a and future observation z^a

$$b^a[X_{k+1}] \doteq \mathbb{P}(X_{k+1} \mid a_{0:k-1}, z_{0:k}, a, z^a) \tag{4}$$

A recursive formula can then be derived

$$b^a[X_{k+1}] = \eta \cdot b[X_k]\mathbb{P}(x_{k+1} \mid x_k, a)\mathbb{P}(z^a \mid X_{k+1}^o), \tag{5}$$

where η is some normalization constant.

From here we can also derive an update rule for the posterior information matrix of the future belief (see e.g. [5]):

$$\Lambda_a^+ = \Lambda + G^T W_{k+1}^{-1} G + H^T V_{k+1}^{-1} H, \tag{6}$$

such that $b[X_{k+1}] = \mathcal{N}(\times, (\Lambda_a^+)^{-1})$. Here, Λ is the original information matrix of the belief $b[X_k]$, and the matrices G and H are the Jacobians $G = \nabla g$ and $H = \nabla h$.

The expression (6) can be rearranged into a more compact form (see [5] for details):

$$\Lambda_a^+ = \Lambda + A^T A, \tag{7}$$

where the collective Jacobian A encapsulates information regarding the models of both the transition and its following observation, in relation to an action a. Each action can be described using a Jacobian of this form, to be used when updating a belief according to Eqs. (5) and (7). Note that A, and hence Λ_a^+, are not dependent on the actual unknown future observation z^a, but on its known model [5].

One can also consider a non-myopic setting, where a represents a sequence of actions $a \doteq \{a_k, \ldots, a_{k+l-1}\}$. The corresponding belief $b^a[X_{k+l}]$ can be written, similarly to Eq. (5), as

$$b^a[X_{k+l}] = \eta b[X_k] \prod_{i=k+1}^{k+l} \mathbb{P}(x_i \mid x_{i-1}, a_{i-1})\mathbb{P}(z_i^a \mid X_i^o), \tag{8}$$

where $z^a \doteq \{z_{k+1}^a, \ldots, z_{k+l}^a\}$.

From here on the time index will be dropped to avoid clutter.

A. Decision Making Under Uncertainty

Given a set of candidate actions $\mathcal{A}$ and a revenue (or objective) function J, the decision making problem is defined as

$$a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, J(b, a), \tag{9}$$

where $b \doteq b[X_k]$ is the current belief. In this context, we wish to measure the predicted uncertainty in future beliefs, and to minimize it. We use entropy as a measure to uncertainty, as commonly used in information-theoretic decision making. In the case of a *Gaussian* belief b, over a vector of size n and a covariance matrix Σ, the entropy is:

$$\textit{entropy}(b) = \frac{1}{2} \ln\left[(2\pi e)^n \det(\Sigma)\right] \tag{10}$$

Consider b to be the current belief and b^a the updated belief after performing an action a and taking the respective observation. In order to minimize the uncertainty in the future belief, or equivalently, to maximize the information gain inducted by the candidate action, we can define the following revenue function for our decision making process:

$$J(b, a) \doteq \frac{1}{|\Sigma_a^+|} = |\Lambda_a^+| = |\Lambda + A^T A|, \tag{11}$$

where Λ, Λ_a^+ are the information matrices of b, b^a, respectively. Looking for an action $a \in \mathcal{A}$ that maximizes $J(b, a)$. In order to examine the revenue of several different actions, using the the information matrix, the inverse of the covariance matrix, is often preferred. The information matrix holds the lucrative property of additivity. Meaning, incorporating new information in order to update the belief, is done by adding it to the current information matrix, as shown in Eq. (7). This property allows us to examine many candidate actions easily. Moreover, information matrices are usually much more sparse to begin with, than covariance matrices.

An alternative way to calculate the same revenue function (11) is by using the triangular square root information matrix, given by Cholesky decomposition:

$$\Lambda = R^T R. \tag{12}$$

The posterior square root information matrix R_a^+ that corresponds to action a such that

$$\Lambda_a^+ = {R_a^+}^T R_a^+ \tag{13}$$

can be efficiently calculated, while exploiting sparsity, e.g. using Givens rotations.

In this form, the determinant (11) can be calculated in linear time - using the diagonal of the triangular matrix. When using this alternative, we move the more significant computational cost from the determinant calculation to the root update, and lose the addetivity of the information in favor of an easier determinant calculation.

B. Objective

In contrast to inference, where, given action and observations, the posterior is calculated only once, information-theoretic decision making requires to evaluate the revenue function for each action, thereby involving numerous such

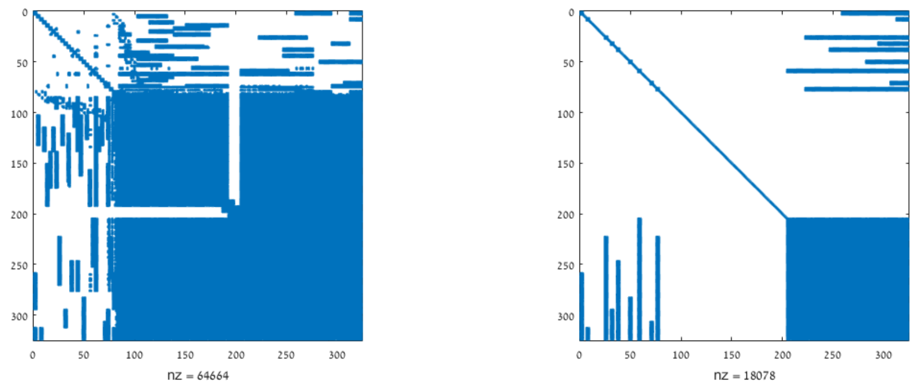


Fig. 1: On the left - the original information matrix taken from an iteration of the SLAM scenario. The state vector holds all previous poses and all the observed landmarks. The sparser part of the matrix is of the loosly correlated poses, which are the first variables in the vector (as expected in an information matrix). The denser part is of the highly correlated landmarks. On the right - its sparse version, generated using algorithm 1. Note the significant difference in the number of non zero elements.

calculations. In particular, this involves determinant calculation of appropriate posterior information matrices, while calculation of a single determinant is by itself $O(n^3)$, in the general case, where n is the state dimensionality. Problems in the real world often encapsulate very large state vectors and groups of candidates actions. Hence, the computational cost of making an uncertainty-related decision can be significant.

The cost of calculation of the determinant is directly dependent on the number of non-zero elements in the matrix, making it significantly lower in sparse matrices. Thus, if we could cut down the number of non-zero elements, and make the matrix sparser, this calculation would be computationally cheaper. Obviously changing a matrix can affect its determinant value and by such, the action selection. Our objective is to find a method to sparsify each Λ_a^+, yet to keep a minimal influence on their determinants (the revenues), and thus to reduce the computational cost needed to make a decision. If discussing the alternative square root matrix form, as seen in Eq. (13), we can equivalently seek a method to sparsify R, in order to improve the update process into R^+. In the next section we provide a formal definition to this objective.

It should be clarified this sparsification is only discussed here in the context of decision making, i.e. action selection. The inference process is not affected and stays exact.

III. Approach

As seen in the revenue function (11), due to the usage of the information form, the original information matrix Λ is involved in the calculation of all future posterior information matrices. Similarly to [4], we propose to find a sparse version of Λ and use it in the information update phase. This will lead to sparser posterior information matrices for all candidate actions. Afterwards, the revenue of each action can be derived in the usual manner, but using these sparse matrices. It also means sparsifying the root matrix R.

Moreover, we wish for this sparsification to have a minimal effect over the action selection. The following set of definitions makes it possible to formally compare this aspect. First, in order to be able to compare beliefs, we define a new kind of metric, in the context of decision making.

Definition 1: Consider a group of actions $\mathcal{A}$, a revenue function J and two beliefs b and b_s (these notations will also be relevant for the definitions to follow).
The ***revenue offset of an action*** a is defined as:

$$\gamma(b, b_s, a) \doteq |J(b, a) - J(b_s, a)|. \tag{14}$$

The ***revenue offset between the two beliefs*** is defined as:

$$\gamma(b, b_s) \doteq \max_{a \in \mathcal{A}} \gamma(b, b_s, a). \tag{15}$$

Definition 2: Two beliefs b, b_s are ***action-consistent***, and marked $b \sim_J^{\mathcal{A}} b_s$, if the following applies $\forall a_{i,j} \in \mathcal{A}$:

$$J(b, a_i) < J(b, a_j) \iff J(b_s, a_i) < J(b_s, a_j) \tag{16}$$

$$J(b, a_i) = J(b, a_j) \iff J(b_s, a_i) = J(b_s, a_j) \tag{17}$$

Meaning, the order of actions in terms of revenue is kept whether starting from b or b_s. Therefore, deriving the best candidate action is equivalent in both cases. Now we can state our objective formally: Given a belief b, we wish to find a sparse and action-consistent b_s.

Note that when the metric between two beliefs is zero, they are action consistent. This rule is given directly from these definitions.

Theorem 1:

$$\gamma(b, b_s) = 0 \quad \Rightarrow \quad b \sim_J^{\mathcal{A}} b_S \tag{18}$$

Proof:

$$\gamma(b, b_s) = 0 \quad \Rightarrow \quad \forall a \in \mathcal{A} \quad J(b, a) = J(b_s, a)$$

$$\Rightarrow \quad J(b, a_i) = J(b_s, a_i),\ J(b, a_j) = J(b_s, a_j)$$

Meaning:

$$J(b, a_i) < J(b, a_j) \iff J(b_s, a_i) < J(b_s, a_j)$$

$$J(b, a_i) = J(b, a_j) \iff J(b_s, a_i) = J(b_s, a_j)$$

■

Keeping this condition not only means the beliefs are action-consistent (maintain the *order* of actions), but also that the *values* of the revenues do not change.

The belief can be sparsified by identifying *uninvolved* variables. Considering a given action, variables in the state vector are *involved* if they are directly updated by the action. Practically, in the collective Jacobian of the action, each of the columns corresponds to a variable of the state vector. A variable is involved if at least one of the entries in its matching column is non-zero. Uninvolved variables correspond to columns of zeros. To clarify, this identification of uninvolved variables is done for each action independently.

```
1  Inputs:
2      A belief b ~ N(x, Λ^{-1})
3      A list of the variables which are uninvolved for all
         candidate actions in A
4  Output:
5      A sparse belief b_s such that γ(b, b_s) = 0

6  Use Cholesky decomposition to find R such that Λ = R^T R
7  Calculate M = R^{-1}
8  Generate a sparse M_s according to:
```

$$(M_s)_{ij} = \begin{cases} M_{ii} & i = j \\ M_{ij} & i \neq j \text{ and the } i\text{-th variable is involved} \\ 0 & i \neq j \text{ and the } i\text{-th variable is never involved} \end{cases}$$

```
9  Calculate R_s = M_s^{-1}
10 Calculate Λ_s = R_s^T R_s
11 return b_s ~ N(x, Λ_s^{-1})
```

Algorithm 1: Sparsification of the belief

Algorithm 1 summarizes the approach for generation of a sparse version of the prior information matrix Λ. The algorithm considers the variables which are uninvolved in all the candidate actions. This allows only a single sparsification process per decision, while keeping consistency for all actions. It is possible to generate a sparse prior approximation per action, or per subgroups of actions, considering the uninvolved variables in each subgroup independently. This will result in a better, more adapted sparsification for the subgroup, since less variables should be involved in this case. Yet, calculation of the sparsification itself has a cost. Finding the optimal number of subgroups (ranging from 1 to the number of actions), to achieve the best performance, is a challenging problem on its own. Here we examine the simple and most general case, which is necessary for assuring the condition is kept for all actions, using a single run of the algorithm.

Theorem 2: Given a belief b, Algorithm 1 yields a belief b_s such that $\gamma(b, b_s) = 0$.

Proof: Consider the Jacboian $A \in \mathbb{R}^{m \times n}$ consists of several row vectors $v_i \in \mathbb{R}^{1 \times n}$, $1 \leq i \leq m$, i.e.

$$A = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \tag{19}$$

It is not difficult to show that $A^T A$ from the information update (7) can be written in additive form in terms of individual vectors v_i, i.e.: $A^T A = \sum_i v_i^T v_i$.

Below we prove Theorem 2 considering single-lined Jacobians $A = v$. The relation above makes the proof valid for the general case (again, due to the additivity in the information space).

From Algorithm 1, we know $\Lambda = R^T R$, $M = R^{-1}$. Thus:

$$\Lambda^{-1} = (R^T R)^{-1} = R^{-1}(R^T)^{-1} = R^{-1}(R^{-1})^T = M M^T$$

The matrix M_s is upper triangular and has the same diagonal as in M (see line 8 in Algorithm 1). Therefore:

$$|M|^2 = |M_s|^2$$

$$|\Lambda| = \frac{1}{|\Lambda^{-1}|} = \frac{1}{|MM^T|} = \frac{1}{|M|^2} = \frac{1}{|M_s|^2} = \frac{1}{|M_s M_s^T|} = \frac{1}{|\Lambda_s^{-1}|} = |\Lambda_s|$$

If we notate $\epsilon = |\Lambda| - |\Lambda_s|$, then in this case $\epsilon = 0$.

Let us choose an action $a \in \mathcal{A}$ with a matching Jacobian vector v. Now:

$$\gamma(b, b_s, a) =$$
$$|\ |\Lambda^+| - |\Lambda_s^+|\ | =$$
$$|\ |\Lambda + vv^T| - |\Lambda_s + vv^T|\ | =$$
(according to the matrix determinant lemma)
$$|\ (1 + v^T \Lambda^{-1} v) \cdot |\Lambda| - (1 + v^T \Lambda_s^{-1} v) \cdot |\Lambda_s|\ | =$$
$$|\ (|\Lambda| - |\Lambda_s|) + v^T \cdot (\Lambda^{-1}|\Lambda| - \Lambda_s^{-1}|\Lambda_s|) \cdot v\ | =$$
$$|\ \epsilon + v^T \cdot [(\Lambda^{-1} - \Lambda_s^{-1}) \cdot |\Lambda| + \epsilon \cdot \Lambda_s^{-1}] \cdot v\ | =$$
$$\eta \cdot |\ v^T \cdot (\Lambda^{-1} - \Lambda_s^{-1}) \cdot v\ | =$$
$$\eta \cdot |\ v^T \cdot (MM^T - M_s M_s^T) \cdot v\ | =$$
$$\eta \cdot |\ \sum_{i=1}^{n} \sum_{j=1}^{n} v_i \cdot (MM^T - M_s M_s^T)_{ij} \cdot v_j\ |$$

If $v_i = 0$ or $v_j = 0$ the inner term is equal to zero.
If $v_i, v_j \neq 0$:

$$(MM^T - M_s M_s^T)_{ij} = (MM^T)_{ij} - (M_s M_s^T)_{ij} =$$
$$\sum_{k=1}^{n} M_{ik} M_{jk} - \sum_{k=1}^{n} M_{sik} M_{sjk} =$$
$$\sum_{k=1}^{n} M_{ik} M_{jk} - \sum_{k=1}^{n} M_{ik} M_{jk} = 0$$

Thus making the inner term zero anyway.

$$\Rightarrow \gamma(p, q, a) = 0$$

■

In an equivalent manner, using the method can be looked at as if the derivation of the best action is based on an adapted version of the revenue function, which uses the sparse version of the initial belief. This function is easier to calculate than the original, but without any impact on the results. Note that only a single run of the sparsification algorithm is required in order to examine any number of actions, since the same sparse initial belief is used in the calculation of all revenues.

$$J_s(b, a) \doteq J(b_s, a) = |\Lambda_s + A^T A| = |\Lambda_{a\ s}^+| = |\Lambda_a^+| \tag{20}$$

An important observation is that according to this scheme, it is also possible to apply the approach to non-myopic decision making process. The same sparsification method can be applied when examining several look-ahead steps at a time, as described in Eq. 8.

The diagram in figure 2 concludes the usage paradigm of the optimization method.

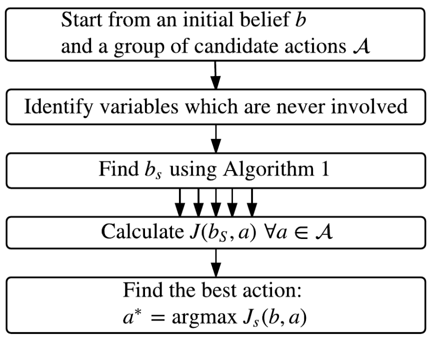


Fig. 2: High-level usage paradigm of the suggested approach.

IV. Results

In this demonstration we wish to prove the improvement in runtime in an actual decision making process, in a simulated SLAM scenario. The simulation consists of a robot navigating in an unknown environment, in which random landmarks are scattered. In the scenario the robot tries to navigate through several predefined world points. The robot aims to navigate between these goals in a *safe* way. Meaning, keeping the uncertainty low throughout the trajectory, by preferring more informative actions. The state vector maintains the entire trajectory X_k and positions of observed landmarks. The robot iteratively decides what is the best future action, executes it, and takes an observation of the environment.

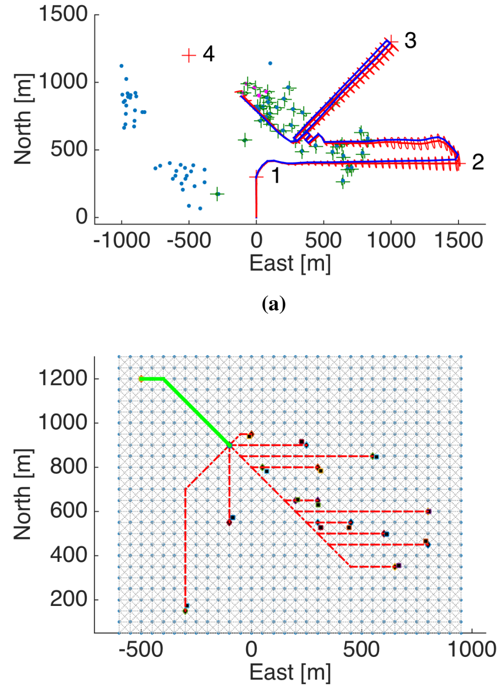
Candidate actions are generated dynamically in every iteration. The actions refer to taking possible short paths around the robot, either to observed close by landmarks (can reduce uncertainty by observing loop closures), or directly to one of the goal points. The actions actually represent several future steps, depending on the length of the suggested trajectory. As previously stated (8), our sparsification method is still applicable in this case.

The total revenue function by which the actions are chosen is of the following form:

$$J(b, a) \doteq w_1 \cdot |\Lambda_a^+| + w_2 \cdot |x_{k+1} - Goal| + w_3 \cdot Penalty(a), \quad (21)$$

where w_1, w_2, $w_3 \in \mathbb{R}$. Our optimization is only relevant to the calculation of the first element - the uncertainty. The other two elements represent the distance to the next goal and penalty on the cost of locomotion (taking a shorter path is preferable). We can use the adapted revenue function (20) to calculate the value of the uncertainty element, as we know it isn't affected by the sparsification.

To test the approach, in each iteration, as a part of the calculation of the overall revenue function (21), we calculate the uncertainty using both the original and adapted revenue functions, J and J_s (11, 20). The revenue is calculated for each candidate action, in order to select the best one. We measure the the total revenue calculation time per iteration for both the original and optimized methods, and compare between the two. The optimized method requires a one time calculation of the sparsification per iteration. It has also been measured and taken into consideration for the optimized method. Overall, in each iteration we compare the time it takes to make the decision in two ways. Note that as the revenue values remain the same, the decision making itself is the same for both methods. The other two elements in Eq. 21 do not change.



(b)

Fig. 3: (a) The 2-dimensional navigation scenario from a top view. The robot navigates between goals 1-4. The red line indicates the estimated trajectory, with the uncertainty ellipses drawn at each state. The blue line indicates the ground truth that the robot passes. Blue dots are landmarks - when observed they are marked green. Note the reduction in size of the red ellipses when observing more landmarks. (b) A demonstration of a single decision. The robot is in the middle point. Comparing candidate trajectories to close by points (red lines). The selected trajectory is highlighted in green. For in depth explanation, see [7], [5]

The accumulation of the measured time throughout the navigation process is shown in Fig. 4. The graph clearly shows a significant improvement in runtime over the original version. It also shows how insignificant the sparsification becomes as the state grows.

Fig. 5 shows the correlation between the percentage of uninvolved variables for a certain decision, to the improvement in runtime which was achieved for that decision. More uninvolved variables translates to a more significant sparsification, and thus to a more significant improvement to the computational cost. As the navigation state vector grows, a higher percentage of the variables becomes uninvolved.

V. Conclusions

This paper introduced a novel optimized approach for decision making under uncertainty. While related optimiza-

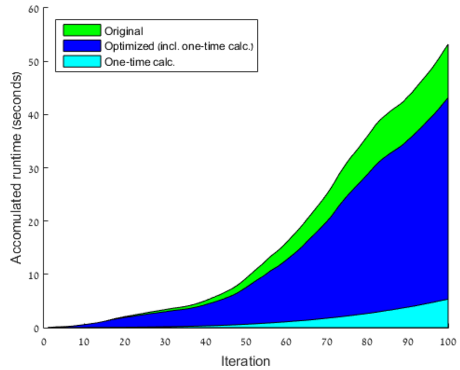


Fig. 4: Comparison of the accumulated decision making time throughout process. In cyan - time to calculate the sparsification (Alg. 1), once for every decision. In blue - the sparcification time, together with the time to calculate all the revenues according to the sparsified information matrix. In green - time to calculate all the revenues with the original information matrix, i.e. without using our method. Note the growing gap in favor of our optimized version, even with the added overhead calculation of the sparsification, runtime is still improving, showing that about 20% of the decision making time could be saved by the 100-th iteration.

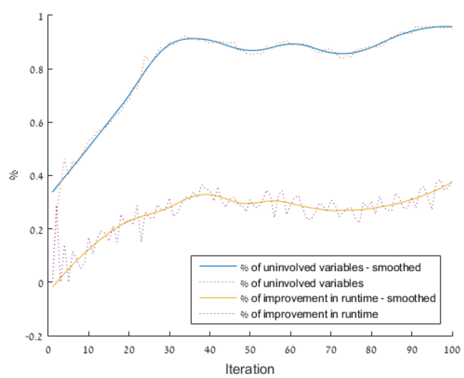


Fig. 5: Correlation between the percentage of uninvolved variables in the state vector, and the improvement in runtime.

tion methods usually focus on sparsifying information in the inference process, here we wished to do so only in the context of decision making. Thus, leaving the inference process intact. This concept also makes the method generic, with no limitations to specific problems.

We used a new sparsification algorithm for the information matrix, in order to reduce the computational cost of the revenue calculation, and making an uncertainty based decision. Not only the sparsification optimizes the performance, it is also proved that it has no effect over the action selection. The benefits of the method has been demonstrated in a simulated SLAM problem. Showing significant improvement in runtime, even when considering the calculation for the sparsification itself. Keeping the same quality of solution while reducing the computational cost makes the approach highly worthwhile.

This concept is a step forward from the current state of the art, and opens the door to an unexplored field of research - optimizations in decision making. Although the method demonstrated in this paper assumes Gaussian beliefs, the leading definitions and the objective - finding an action consistent and sparse approximation by minimizing the revenue offset, are valid as a general concept for reducing computational complexity and achieving efficient decision making. This objective can be further examined with less restrictive conditions. It can also be used for a non-information related decision making. The supporting definitions presented here, creates a well defined environment for future progressions. Possible directions can examine more ways to preserve action-consistency, or even discuss a deviation to the non-exact case.

References

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

[2] Nicholas Carlevaris-Bianco, Michael Kaess, and Ryan M Eustice. Generic node removal for factor-graph slam. *IEEE Trans. Robotics*, 30(6):1371–1385, 2014.

[3] G. Huang, M. Kaess, and J.J. Leonard. Consistent sparsification for graph optimization. In *Proc. of the European Conference on Mobile Robots (ECMR)*, pages 150 – 157, 2012.

[4] V. Indelman. No correlations involved: Decision making under uncertainty in a conservative sparse information space. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):407–414, 2016.

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

[6] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31:217–236, Feb 2012.

[7] A. Kim and R. M. Eustice. Active visual slam for robotic area coverage: Theory and experiment. *Intl. J. of Robotics Research*, 2014.

[8] A. Kim and R.M. Eustice. Perception-driven navigation: Active visual SLAM for robotic area coverage. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3196–3203, 2013.

[9] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. of Machine Learning Research*, 9:235–284, 2008.

[10] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[11] C. Leung, S. Huang, and G. Dissanayake. Active SLAM using model predictive control and attractor based exploration, 2006.

[12] Mladen Mazuran, Gian Diego Tipaldi, Luciano Spinello, and Wolfram Burgard. Nonlinear graph sparsification for slam. In *Robotics: Science and Systems (RSS)*, pages 1–8, 2014.

[13] S. Patil, G. Kahn, M. Laskey, J. Schulman, K. Goldberg, and P. Abbeel. Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation. In *Intl. Workshop on the Algorithmic Foundations of Robotics*, 2014.

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

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

[16] A. Singh, A. Krause, C. Guestrin, and W.J. Kaiser. Efficient informative sensing using multiple robots. *J. of Artificial Intelligence Research*, 34:707–755, 2009.

[17] W. Sun, S. Patil, and R. Alterovitz. High-frequency replanning under uncertainty using parallel sampling-based motion planning. *IEEE Trans. Robotics*, 31(1):104–116, 2015.

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

[19] J. Vial, H. Durrant-Whyte, and T. Bailey. Conservative sparsification for efficient and consistent approximate estimation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 886–893. IEEE, 2011.

[20] J. M. Walls, S. M. Chaves, E. Galceran, and R. M. Eustice. Belief space planning for underwater cooperative localization. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015.