Scalable 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 sparsification method for efficient decision making under uncertainty and belief space planning in high dimensional state spaces. By using a sparse version of the state's information matrix, we are able to improve the high computational cost of examination of all candidate actions. We also present an in-depth analysis for the general case of approximated decision making, and use it in order to set bounds over the induced error in potential revenue. The scalability of the method allows balancing between the degree of sparsification and the tolerance for this error, in order to maximize its benefits. The approach differs from recent methods by focusing on improving the decision making process directly, and not as a byproduct of a sparsification of the state inference. Eventually, we demonstrate the superiority of the approach in a SLAM simulation, where we manage to maintain the accuracy of the solution, while demonstrating a significant improvement in run time.**

I. Introduction

Solving a decision making problem is the essence of any intelligent autonomous agent. The objective of this optimization problem is finding the most beneficial action, in relation to some measure, or a revenue function. Robots which are set in the real world are often required to account for its uncertainty when making decisions, in order to provide reliable and robust results. There are multiple possible sources for this uncertainty, e.g. a dynamic environment in which unpredictable events might occur; Noisy or limited observations, such as a limited camera range and an inaccurate GPS signal; and inaccurate delivery of actions.

The notions of decision making under uncertainty and belief space planning (BSP) are applicable to the solution of numerous problems. These include simultaneous localization and mapping (SLAM) (e.g. [7], [5], [10]), sensor deployment and active sensing (e.g. [8], [14]), and in recent years even more profound problems such as natural language processing (NLP) (e.g. [13]).

Moreover, long-term autonomous navigation, sensor deployment over large areas, and any kind of problem in which a state is described with numerous features, often require dealing with large states as well. These settings are translated to high dimensional probabilistic states, known commonly as beliefs (as in BSP, see [12]). In this case, the revenue function account for the uncertainty of belief, yet uncertainty measures, such as entropy, are expensive to calculate. Consider the fact that making a decision in a naive way requires calculation of this measure for every candidate action, in a possibly very large group of actions. Overall the total computational cost of the problem can turn exceptionally expensive, thus making the problem challenging for online systems, or when having a limited processing power.

Recently several approaches have been introduced in order to improve the high computational cost of dealing with the uncertainty, often with some penalty on the accuracy of the results. Many approaches ignore the least significant nodes in the underlying factor graph (e.g. [1], [3], [11], [15]), in order to achieve long-term autonomy. In the context of SLAM problems, some try exploit the sparsity of the information, or reuse previous calculations (e.g. [6], [9]). These approaches have a few common downsides. First, the sparsification is done in the context of inference, which over time causes an unnecessary deviation from the true state. Second, although some of these approaches demonstrate good performance, the induced error from using them is not modeled nor bounded, and therefore the results can not be guaranteed. In our approach we try to take care of both issues.

We are looking for a computationally efficient method to find the best action, while sparing the naive explicit calculation of all revenues. [4] introduced a first attempt to try and separate the approximation of the decision making from the inference. That approach ignores the correlations between variables, only for the action selection, and achieves a great improvement in runtime. Yet, the concept was preliminary and assumed several limitations, such as unary update model, and a greedy selection. The idea was then thoroughly examined in [2]. It introduced new ideas for comparison of states in the context of decision making, which are later used to find a sparsification method which yields action consistent results, i.e. with no effect over the action selection.

In this follow-up paper we extend this analysis and explore inexact approximations of the decision making, and their influence on the results, in order to set bounds on the induced error. This qualitative analysis is relevant to any decision making process, and not only in an information-related context, making it a good foundation for a yet unexplored research area of decision making approximations. Using this analysis, we then introduce a scalable sparsification method, in the context of decision making under uncertainty, and bound the error it causes. The benefits of the approach are finally demonstrated in a SLAM simulation, where a significant improvement in run time is achieved.

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.

II. Problem Formulation

A. Decision Making Under Uncertainty

Consider an iterative Markovian state inference process, in which every update iteration consists of taking an observation, and then performing a certain action, with the following transition and observation models

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

both containing some *Gaussian* noise. These models can be described probabilistically as $\mathbb{P}(x_{k+1} \mid x_k, a)$ and $\mathbb{P}(z_k \mid X_k)$.

We can discuss the posterior distribution at time k over the state vector X_k

$$b_k \doteq \mathbb{P}(X_k \mid a_{0:k-1}, z_{0:k}), \tag{3}$$

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. To describe such a distribution we use the term *belief*. The variables in the state vector are defined with relation to the discussed problem, e.g. for SLAM problems it can include previous poses of the agent and the locations of observed landmarks.

Considering a basic belief $b_k = \mathcal{N}(X_k, \Lambda_k^{-1})$. An update rule for the information matrix Λ of this belief, after performing the next action a_k, and taking the following observation z_{k+1}, can be derived (see [2])

$$\Lambda_{k+1} = \Lambda + G^T W_{k+1}^{-1} G + H^T V_{k+1}^{-1} H, \tag{4}$$

such that $b_{k+1} = \mathcal{N}(\times, \Lambda_{k+1}^{-1})$. The matrices G and H are the Jacobians $G = \nabla g$ and $H = \nabla h$.

This expression (Eq. 4) can be rearranged into a more compact form (see [5] for details):

$$\Lambda_{k+1} = \Lambda + A_k^T A_k \tag{5}$$

where the collective Jacobian A_k encapsulates information regarding the models of both the action a_k and its following observation z_{k+1}. Each update iteration can be described using a Jacobian of this form, in relation to the performed action. Note that A_k, and hence the information matrix Λ_{k+1}, are not dependent on the actual unknown future observation z_{k+1}, nor on the result of performing the action, but only on their given models.

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

$$a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, J(b, a), \tag{6}$$

In this context, we wish to minimize the uncertainty in the future belief, or equivalently maximize the information gain. We use entropy as a measure to uncertainty, as commonly used in information-theoretic decision making. The entropy of a *Gaussian* belief b, over a vector of size n, with an information matrix Λ is:

$$\textit{entropy}(b) = \frac{1}{2} \ln\left[(2\pi e)^n \det(\Lambda^{-1})\right] \tag{7}$$

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 entropy of b^a, we can define the following revenue function:

$$J(b, a) \doteq |\Lambda^a| = |\Lambda + A^T A|, \tag{8}$$

where Λ, Λ^a are the information matrices of b, b^a, respectively, and A is the collective Jacobian of a.

Note that incorporating new information in order to update the belief, is done by adding it to the current information matrix, as shown in Eq. (5). This property allows us to examine many candidate actions, and their posterior information, easily.

Calculation of the revenues requires to compute a determinant of the posterior information for every candidate action. A single determinant computation of a matrix of size $n \times n$ is valued at $O(n^3)$ at worst. Obviously the sparser the matrix is, the less expensive it is to calculate this value. For this reason, sparsifying all posterior information matrices, would essentially reduce the computational cost of the problem. Since the same basic information matrix is a factor in all those matrices, sparsifying it only, would inherently mean sparsifying all posterior information matrices, as needed. Yet, this sparsification can affect the revenues. Optimally this effect should be minimal, in order to maintain the original action selection.

B. Objective

[2] introduced the concept of action consistency. Two states are action consistent if when performing the group of actions on them, the trend, or order of the actions is kept (in relation to the revenue in their posterior belief). Let us recall that definition. Previously stated specifically for belief space planning, but it can be revised to be relevant to any kind of decision making process, even with non-probabilistic states.

Definition 1: Consider a group of actions $\mathcal{A}$ and a revenue function $J(state, action)$ (these notations will also be relevant for the definitions to follow). Two states b, b_s are ***action consistent***, in relation to J and $\mathcal{A}$, and marked $b \sim b_s$, if the following applies $\forall a_{i,j} \in \mathcal{A}$:

$$J(b, a_i) = J(b, a_j) \iff J(b_s, a_i) = J(b_s, a_j) \tag{9}$$
$$J(b, a_i) < J(b, a_j) \iff J(b_s, a_i) < J(b_s, a_j) \tag{10}$$

This situation represents a tight correlation between the two states, such that action selection, starting from either, is equivalent. The revenue offset was offered as a "metric" between states. It yields an easy to prove condition for action consistency, when the offset is zero (this is a sufficient condition only).

Definition 2: Consider two states b and b_s. The ***revenue offset of an action*** a is defined as:

$$\gamma(b, b_s, a) \doteq |J(b, a) - J(b_s, a)| \tag{11}$$

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

$$\gamma(b, b_s) \doteq \max_{a \in \mathcal{A}} \gamma(b, b_s, a) \tag{12}$$

In [2] the revenue offset was used to find a sparsification method which yields a sparse and action consistent approximation of the state. This approximation was then used as the basic value for the decision making iteration, in order to reduce the computational cost. As stated, this had no effect over the action selection and the results.

We now wish to examine what happens to the correlation between the states when allowing a looser coupling and a non-zero revenue offset. By exploiting this ease of restrictions, we wish to find an improved sparsification method. In this scenario, an error, or loss of revenue is expected to be caused by using the approximated state for the decision making. To provide valuable results, it is important to set bounds over this error.

III. Approach

A. Bounding the Error

As stated, it is already known that when the revenue offset between two states is zero, the states are action consistent, and the action selection in this case is equivalent from both. Considering a state b, over which we wish to examine a group of actions $\mathcal{A}$; b_s is a given approximation of that state. Solving the decision making problem means finding the action that yield the maximum revenue. When using the approximation b_s as the base state, instead of b, the revenues of the actions might change, thus the chosen best action is not necessarily the actual best action, when applied on the real state b. This difference of the theoretically maximum revenue and lesser generated revenue, is the *error* we bare for using the approximation.

Formally, for

$$\begin{aligned} a^* &= \underset{a}{\operatorname{argmax}}\, J(b,a) \\ a_s^* &= \underset{a}{\operatorname{argmax}}\, J(b_s,a), \end{aligned} \tag{13}$$

the error is $J(b,a^*) - J(b,a_s^*)$. We wish to bound and control this possible error that is induced from using the approximation. To do so we can use the revenue offset between b and b_s. First, we prove the following supporting theorem:

Theorem 1:

$$|J(b,a^*) - J(b_s,a_s^*)| \leq \gamma(b,b_s) \tag{14}$$

Proof:

Considering the false assumption:

$$|J(b,a^*) - J(b_s,a_s^*)| > \gamma(b,b_s) \doteq \max_{c\in A} |J(b,c) - J(b_s,c)|$$

If $J(b,a^*) \geq J(b_s,a_s^*)$:

$$\begin{gathered} |J(b,a^*) - J(b_s,a_s^*)| > |J(b,a^*) - J(b_s,a^*)| \\ J(b,a^*) - J(b_s,a_s^*) > J(b,a^*) - J(b_s,a^*) \\ J(b_s,a^*) > J(b_s,a_s^*) \end{gathered}$$

And this is a contradiction to the definition of a_s^*.

If $J(b,a^*) < J(b_s,a_s^*)$:

$$\begin{gathered} |J(b,a^*) - J(b_s,a_s^*)| > |J(b,a_s^*) - J(b_s,a_s^*)| \\ J(b_s,a_s^*) - J(p,a^*) > J(b_s,a_s^*) - J(b,a_s^*) \\ J(b,a_s^*) > J(b,a^*) \end{gathered}$$

And this is a contradiction to the definition of a^*. In any way our false assumption is not possible. Therefore:

$$|J(b,a^*) - J(b_s,a_s^*)| \leq \gamma(b,b_s)$$

■

And now we can conclude the following error bound:

Theorem 2:

$$0 \leq J(b,a^*) - J(b,a_s^*) \leq 2\cdot\gamma(b,b_s) \tag{15}$$

Proof:

The following is given directly from the definition of a^*:

$$0 \leq J(b,a^*) - J(b,a_s^*)$$

Considering the false assumption:

$$J(b,a^*) - J(b,a_s^*) > 2\cdot\gamma(b,b_s)$$

Using the supporting theorem:

$$\begin{gathered} J(b,a^*) - J(b_s,a_s^*) \leq |J(b,a^*) - J(b_s,a_s^*)| \leq \gamma(b,b_s) \\ J(b,a^*) \leq J(b_s,a_s^*) + \gamma(b,b_s) \\ J(b_s,a_s^*) + \gamma(b,b_s) - J(b,a_s^*) > 2\cdot\gamma(b,b_s) \\ J(b_s,a_s^*) - J(b,a_s^*) > \gamma(b,b_s) \\ \gamma(b,b_s,a_s^*) > \gamma(b,b_s) \end{gathered}$$

And this is a contradiction to the definition of γ. Therefore:

$$J(b,a^*) - J(b,a_s^*) \leq 2\cdot\gamma(b,b_s)$$

■

Meaning, the error of using the approximation for decision making can always be bounded using the offset between the original state and its approximation. Note that important benefit of bounding the error using offset is that it does not require finding the actual best actions a^* and a_s^*, since the bound is absolute and not dependent on them. This allows to decide in advance, before solving the decision making problem, whether the approximation is good enough for us to use within the limits of the error we can bare.

This conclusion is general for every decision making process, and not limited for a specific revenue function nor type of state. It extends the general foundations provided in [2], for a qualitative decision making analysis.

B. Sparsification Method

[2] introduced a consistent sparsification algorithm for the information matrix of a belief b, which was proven to return an approximation b_s s.t. $\gamma(b,b_s) = 0$ and thus being exact and induce no error. According to that algorithm, the belief is sparsified by identifying *uninvolved* variables. Considering a given action, variables in the state vector are *involved* if they are directly updated by applying 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, while uninvolved variables correspond to columns of zeros. The identification of uninvolved variables is done for each action independently, and the algorithm

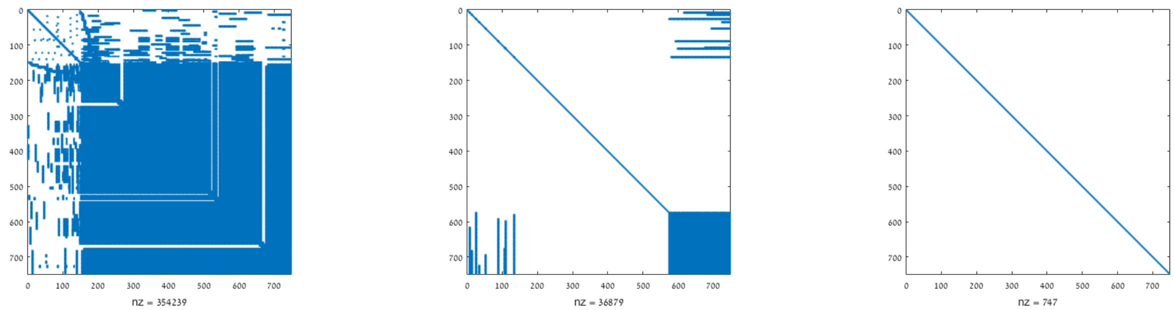

Fig. 1: On the left - the original information matrix taken from an iteration of the SLAM scenario. The state vector holds all previous poses and all the observed landmarks. The sparser part of the matrix is of the loosly correlated poses, which are the first variables in the vector (as expected in an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

considers the variables which are uninvolved for all the candidate actions, thus keeps consistency for all actions, using a single sparsification process per decision.

We now define an extended version, in the form of Algorithm 1, in which we allow removal of elements of involved rows as well. Using Definition 2 and Theorem 2 we can model the error induced by using this approximation for decision making under uncertainty.

Inputs:
A belief $b \sim \mathcal{N}(x, \Lambda^{-1})$
A group of row indexes K
Output:
A sparse approximated belief b_s
Use Cholesky decomposition to find R such that $\Lambda = R^T R$
Calculate $M = R^{-1}$
Generate a sparse M_s according to:

$$(M_s)_{ij} = \begin{cases} M_{ii} & i = j \\ M_{ij} & i \neq j \text{ and } i \notin K \\ 0 & i \neq j \text{ and } i \in K \end{cases}$$

Calculate $R_s = M_s^{-1}$
Calculate $\Lambda_s = R_s^T R_s$
return $b_s \sim \mathcal{N}(x, \Lambda_s^{-1})$

Algorithm 1: Scalable sparsification of a belief

In practice, in every sparsified variable in Algorithm 1, all the elements in its corresponding row in matrix M are zeroed (besides the diagonal). This removal of elements in the matrix M_s bubbles back the information matrix Λ_s and causes it to be more sparse.

We now wish to examine the error induced by using this sparsification. The following analysis considers a single-row collective Jacobian. In principle, since multi-row Jacobians can be represented as a sum of single-row Jacobians, it is possible to extend this work in order to describe the revenue offset of more challenging cases, yet further work is required. It is also easy to show that $\Sigma = \Lambda^{-1} = MM^T$ and $|\Sigma| = |\Sigma_s|$ (see [2] for details). For an action $a \in \mathcal{A}$ with a matching Jacobian vector v:

$$\gamma(b, b_s, a) =$$
$$|\,|\Lambda^a| - |\Lambda_s^a|\,| =$$
$$|\,|\Lambda + vv^T| - |\Lambda_s + vv^T|\,| =$$
(according to the matrix determinant lemma)
$$|\,(1 + v^T\Lambda^{-1}v) \cdot |\Lambda| - (1 + v^T\Lambda_s^{-1}v) \cdot |\Lambda_s|\,| =$$
$$\left| \frac{1}{|\Sigma|}(1 + v^T\Sigma v) - \frac{1}{|\Sigma_s|}(1 + v^T\Sigma_s v) \right| =$$
(Σ is PSD and therefore its determinant is non-negative)
$$\frac{1}{|\Sigma|} \cdot \left|(1 + v^T\Sigma v) - (1 + v^T\Sigma_s v)\right| =$$
$$\frac{1}{|\Sigma|} \cdot \left|v^T(\Sigma - \Sigma_s)v\right| =$$
$$\frac{1}{|\Sigma|} \cdot \left| \sum_{i=1}^{n} \sum_{j=1}^{n} v_i \cdot (\Sigma - \Sigma_s)_{ij} \cdot v_j \right|$$

Now, let us examine the inner addends of this summation:

$$v_i \cdot (\Sigma - \Sigma_s)_{ij} \cdot v_j =$$
$$v_i \cdot (MM^T - M_s M_s^T)_{ij} \cdot v_j =$$
$$v_i v_j \cdot \left((MM^T)_{ij} - (M_s M_s^T)_{ij}\right) =$$
$$v_i v_j \cdot \left(\sum_{k=1}^{n} M_{ik}M_{jk} - \sum_{k=1}^{n} M_{s\,ik}M_{s\,jk} \right)$$

If $i, j \notin K$ then $M_{ik} = M_{s\,ik}$, $M_{jk} = M_{s\,jk}$, and the two sums cancel each other. If $i, j \in K$ then the right sum in zero, and we get:

$$v_i v_j \cdot \Sigma_{ij} \tag{16}$$

If $j \in K$, $i \notin K$ we get:

$$v_i v_j \cdot (\Sigma_{ij} - M_{ij}M_{jj}) \tag{17}$$

If $j \notin K$, $i \in K$ we get:

$$v_i v_j \cdot (\Sigma_{ij} - M_{ii}M_{ji}) \tag{18}$$

After summing up the addends according to these rules, we can derive the final expression for the revenue offset:

$$\gamma(b, b_s, a) =$$
$$\frac{1}{|\Sigma|} \cdot \left| \sum_{i \in K} v_i \cdot \left(\sum_{j \in K} v_j \cdot \Sigma_{ij} + \sum_{j \notin K} v_j \cdot 2(\Sigma_{ij} - M_{ii}M_{ji}) \right) \right| \tag{19}$$

This expression represents the revenue offset *of an action*, and is dependent on the selection of the variables in K. Note that if i is uninvolved (i.e. $v_i = 0$), the corresponding term becomes zero anyway, and does not affect the revenue offset. Thus, the final summation describes penalty over involved variables ($v_i \neq 0$), that are chosen to K. To find the overall revenue offset of an approximation, given a known group K, one should return the maximal value among all actions.

The cost for calculating the expression in Eq. 19, in order to *bound the the error* induced by a *specific* approximation (known K) is $O(\#\ of\ actions \cdot |K| \cdot n)$. Yet, it is possible to reduce this cost by settling for a less tight bound. For example, by using an upper bound for the values of v_i, considering all candidate Jacobians, and calculating a (single) bound over the offset. In this case the cost can be reduced all the way to $O(|K| \cdot n)$. A middle solution can also be choosing those upper bounds for subgroups of similar actions.

As for the sparsification itself, its calculation is done once with no dependence on the number of actions. Inverting the matrix, and computing the Cholesky decomposition are the most dominant calculations of the algorithm, making it $O(n^3)$ at worst. The recalculation of the adapted information matrix becomes easier the more sparse the matrix is. Thus, a higher degree of sparsification - a larger group K - would actually mean a quicker computation.

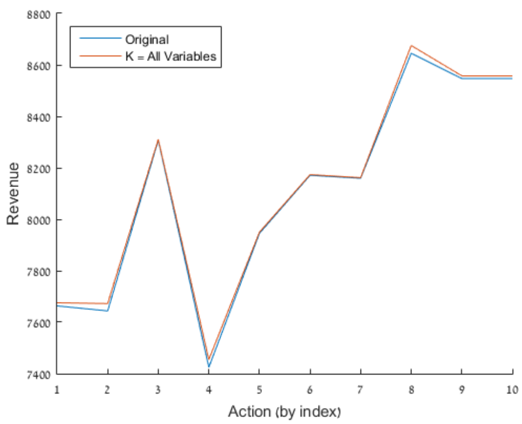

Fig. 2: The revenues of the action which were calculated from the information matrices from Figure 1. Comparing the original version with the sparsified version, for which K includes all variables. It is clearly visible that, despite not being fully guaranteed, the offset between the graphs is slim, and the action selection does not change.

C. Scalability

Every selection of K results in a different approximation of the state, with a different degree of sparsification and a different bound on the error it causes. Assuming it is possible to tolerate a certain degree of inaccuracy in the results, we can analyze whether using a certain approximation can guarantee not exceeding this allowance. Optimally, even actively choosing the most cost-effective group K, that gives the highest degree of sparsification, while exploiting this tolerable error range. For example, a navigating robot that can tolerate a certain level of uncertainty along its trajectory.

A wise selection of K also bares a certain cost, yet it can prove itself profitable by saving more time on the calculation of the revenues, especially for a large group of candidate actions, where the initial investment becomes less significant.

We can scalably choose the rows in K according to some heuristic - adding up rows until the bound no longer guarantees satisfying results. Adding more rows would essentially improve the degree of sparsification. A suggested heuristic for choosing rows for K is the ratio between the number of elements in the row, and the contribution to the error caused by removing this row by itself (a relatively easy calculation). Other options are a simple random selection until the criteria is met, genetic algorithms, and more. This discussion is beyond the scope of this paper, and is considered an optimization problem by itself.

IV. Results

In this simulated SLAM scenario, we wish to demonstrate improvement in performance through the usage of our approach. In order to keep this example easy to follow, the demonstration focuses on presenting the actual runtime improvement achieved by the sparsification, using *predefined* selection of K. We do not try to optimize the group K in attempt to find the most cost effective sparsification.

For comparison we examine side-by-side three different degrees of sparsification, using three options for K: An empty group, i.e. the original version with no sparsification; The group of uninvolved variables, which has no influence on the result, but on the performance only; All variables - the highest degree of sparsification.

The simulation itself 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 in a *safe* way. Meaning, keeping the uncertainty of the state low throughout the trajectory, by preferring more informative actions. The state vector consists of the entire trajectory X_k and positions of observed landmarks. Candidate actions are generated dynamically in every iteration, and represent navigation in short paths around the robot, either towards landmarks (can reduce uncertainty by observing loop closures), or towards a goal points. The robot iteratively decides what is the best future action, executes it, and takes an observation of the environment.

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

$$J(b,a) \doteq w_1 \cdot |\Lambda^a| + w_2 \cdot |x_{k+1} - Goal| + w_3 \cdot Penalty(a), \quad (20)$$

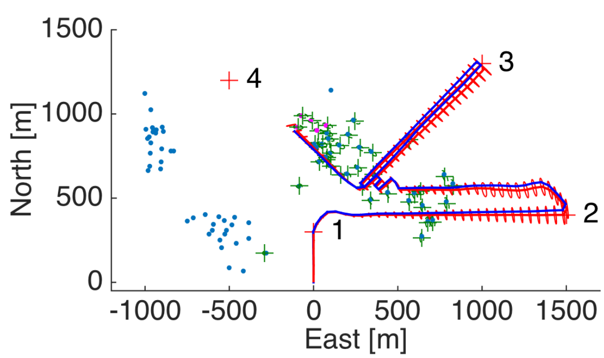

Fig. 3: The 2-dimensional navigation scenario from a top view. The robot navigates between goals 1-4. The red line indicates the estimated trajectory, with the uncertainty ellipses drawn at each state. The blue line indicates the ground truth that the robot passes. Blue dots are landmarks - when observed they are marked green. Note the reduction in size of the red ellipses when observing more landmarks.

where w_1, w_2, $w_3 \in \mathbb{R}$. Our method 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 locomotion (taking a shorter path is preferable).

To test the approach, in each iteration, we calculate the uncertainty in this revenue function using the three versions of the information matrix, according to three sparsification configurations. The revenue is calculated for each candidate action. We measure the total revenue calculation time per iteration, along with the one time calculation of the sparsification itself (for the latter two configurations. Obviously no such calculation for the original configuration). Overall, in each iteration we compare the total decision making time.

It should be noted that the number of candidate actions in each iteration in this scenario is small (averaging at 10). The more candidate actions there are, the less significant the sparsifcation cost becomes in the iteration. Usually in a real scenario, the number of candidate actions is much bigger, making the relative improvement even more profound.

In Figure 2 we compare the revenues of the actions, which were calculated from the information matrices from Figure 1, in that iteration, where for the sparsified version, K includes both the involved and uninvolved variables (the matrix on the right). It is clearly visible that the offset between the graphs is slim. Hence, even though Algorithm 1 does not guarantee a consistent approximation, in practice, the results still maintain accuracy, and the action selection does not change. This situation was repetitive throughout the process.

Figure 4 shows a comparison of the accumulated decision making time, for each of the configurations. The growing saving in run time is clearly visible, and is correlated with the growth of state dimension. It also shows that this difference is more significant for a larger group K - in this case, the one which contains all variables.

V. Conclusions

The work presented in this paper is deviation from the usual approaches which try to improve the high computational cost of decision making under uncertainty, and holds multiple benefits over them. Firstly, we define theoretical foundations for approximation of decision making processes, which is an unexplored and novel concept. Despite their theoretical quality, these ideas prove to have feasible values, as they are then used to define a new approximation method.

This highly scalable method allows to control the degree of sparsification in exchange of lost in revenue from the action selection, which is possible due to bounds we set over the error caused by the approximation. Assuming a certain error is acceptable in the context of the problem, the bound allows us to choose the most cost-effective sparsification which still guarantees results within this acceptable range.

The sparsification method in this paper is just an example in the context of belief space planning. In a similar way, more approximation methods can be developed in other contexts, with other state structures and revenue functions.

Furthermore, a significant improvement in performance has been demonstrated in the SLAM simulation, by using our approach. Thus, showing relevance and possible benefits for online BSP in computationally-constrained robots.

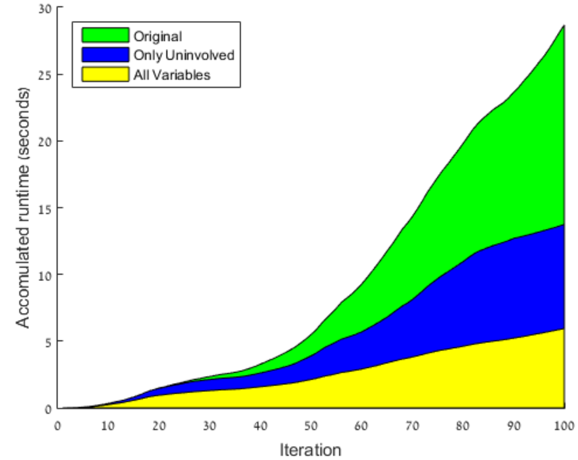

Fig. 4: Comparison of the accumulated decision making time throughout process. In green - calculation with the original information matrix, i.e. without using our method. In blue - sparsification of uninvolved variables. In yellow - a higher degree of sparsification, using the involved variables as well. Note the growing gap between the three configurations.

References

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

[2] K. Elimelech and V. Indelman. Consistent sparsification for efficient decision making under uncertainty in high dimensional state spaces. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017.

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

[8] 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.

[9] 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.

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

[11] 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.

[12] Robert Platt Jr, Russ Tedrake, Leslie Kaelbling, and Tomas Lozano-Perez. Belief space planning assuming maximum likelihood observations. 2010.

[13] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 93–100. Association for Computational Linguistics, 2000.

[14] 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.

[15] 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.