Belief Change Maximisation for Hydrothermal Vent Hunting Using Occupancy Grids

Zeyn Saigol
and Richard Dearden
and Jeremy Wyatt
School of Computer Science
University of Birmingham, UK
Email: {zas, rwd, jlw}@cs.bham.ac.uk

Bramley Murton
National Oceanography Centre
Southampton, UK
Email: bjm@noc.soton.ac.uk

***Abstract*— The problem of where a mobile robot should go to efficiently build a map of its surroundings is frequently addressed using entropy reduction techniques. However, in exploration problems where the goal is to find an object or objects of interest, such techniques can be a useful heuristic but are optimising the wrong quantity. An example of such a problem is an autonomous underwater vehicle (AUV) searching the sea floor for hydrothermal vents. The state of the art in these problems is information lookahead in the action-observation space which is computationally expensive. We present an original belief-maximisation algorithm for this problem, and use a simulation of the AUV problem to show that our method outperforms straightforward entropy reduction and runs much faster than information lookahead while approaching it in terms of performance. We further introduce a heuristic using an orienteering-problem (OP) solver, which improves the performance of both our belief-maximisation algorithm and information lookahead.**

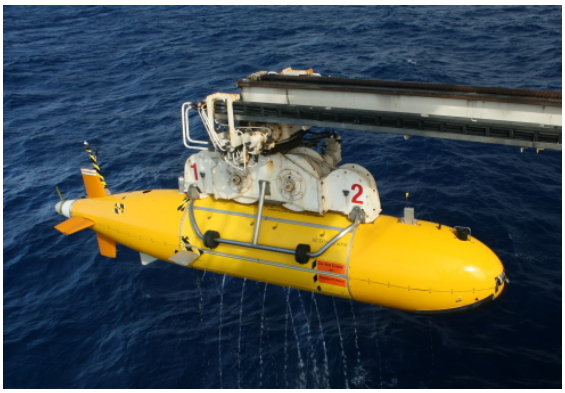

Fig. 1. The Autosub 6000 AUV that was used for vent finding in March 2010.

I. INTRODUCTION

Many applications require a robot to search an unknown environment for an item or items of interest. In such problems the robot has to balance acting to reduce its uncertainty about the environment against acting to directly visit or retrieve the items. We explore this challenge in the domain of an Autonomous Underwater Vehicle (AUV) prospecting for hydrothermal vents, which are superheated outgassings of water found on mid-ocean ridges, and are of great interest to marine scientists.

Hydrothermal vents are formed where continental plates diverge. They can be detected from a distance because they emit a plume that rises several hundred metres above the vent and is spread by diffusion and ocean currents. Detecting a plume gives only limited information about the location of the source vent for several reasons: first, because turbulent flows and changing currents mean there is no steady gradient that can be traced to the vent; second, it is not possible to relate the strength of the plume directly to distance from a vent because some vents are more active than others; finally, vents are often found in clusters and plumes from several vents mix. Current methods for mapping hydrothermal vents employ a series of nested box surveys in which the AUV "mows the lawn" in progressively smaller areas, chosen after analysis of each survey. If the AUV could perform on-board analysis and planning rather than following an exhaustive search pattern it could cover a much larger area while still mapping all or most vents. The domain is described in more detail in [1].

This problem decomposes into two parts, mapping and planning. In mapping the AUV infers a map of likely locations for vents based on the observation history. As stated above this is a challenging problem; however we focus on planning so we adopt the recent vent mapping algorithm of Jakuba [2]. This outputs an occupancy grid (OG) map, containing the probability that there is a vent for each location.

The planning problem is as follows: given such an OG map, choose where to go to localise the vents as efficiently as possible. This planning-for-mapping problem is different from normal indoor mobile robot mapping because the sensor model is non-local and highly uncertain; specifically, unlike sonar or laser sensors, hydrothermal plume detectors do not provide a direct estimate of the distance to a vent, and can be triggered by a very distant vent. In Jakuba's algorithm, the ocean current (measured over the lifetime of the agent) is the key clue used to assign probability to vent locations.

The only actions in this problem are moving in the world. We assume that sensing occurs at every step, returning either that a vent has been found, that the plume from a vent has

been detected, or nothing. The challenge is that we don't just want to visit potential vents given our current probability distribution over their locations, but also to choose actions that give us more information about where the vents might be. This information-gathering with rewards problem is best modelled as a partially observable Markov decision problem (POMDP), but here the state space is far too large to be solved by conventional POMDP methods. The problem is described more formally in Section II.

In this paper, we present two novel approaches; first a pair of algorithms that attempt to maximise the change in the belief state of the agent, and second the calculation of a correction term for any belief-state lookahead method based on the Orienteering Problem (OP). In both cases, the belief state of the agent is represented by the OG.

Our first pair of algorithms use what we call belief change maximisation and were inspired by the deficiencies of entropy based approaches in this domain. To derive them we start by showing how to extend the myopic entropy reduction developed by Vergassola et al [3]—which they term *infotaxis*—from dealing with a single venting source to multiple sources. In so doing the problem becomes similar to entropy reduction approaches used in laser based OG mapping [4]. The difficulty in our domain is that cells in the OG have very low prior probability of occupancy which means entropy often *rises* when we get useful information. The belief change maximisation algorithms are discussed further in Section III, together with results showing that they perform significantly better than both infotaxis and two approaches used on real vehicles: a mow-the-lawn search pattern and a reactive approach similar to chemotaxis [5].

The second contribution is based on the idea of using an orienteering problem (OP) solver [6] to better approximate the value of taking a sequence of actions in the OG. We applied this correction to the belief change maximisation algorithms of Section III, and also to our previous approach for tackling this domain [7], which was based on an online POMDP solver. The OP correction resulted in improved performance in both cases and Section IV gives the algorithm and results.

II. PROBLEM DETAILS

As we said above, we rely on an existing mapping algorithm to compute an OG map based on observations [2]. It is designed specifically for creating a map of vents using plume detection data obtained by an AUV. OGs [8] work by dividing the space into a regular grid and calculating cell occupancy based on a series of sensor readings. In this domain, being occupied means containing a vent. However, classic OG algorithms do not work for vent mapping because they assume that a sensor reading is determined by a single test cell, and is independent of previous sensor readings. In contrast, for vent mapping the sensor can register a plume emitted from any of a large number of neighbouring cells, and previous sensor readings are vital to help narrow down which cells are likely to have been responsible. As Jakuba points out, the errors induced by this classic OG assumption are exacerbated when the prior probability of occupancy is very low (as it will be for vents due to their sparse distribution).

Jakuba develops a more accurate OG approach, firstly by compressing readings from multiple sensors into a binary detection or non-detection of a hydrothermal plume. Given this binary detection input, he is able to develop a tractable approximate algorithm that produces significantly better results in low-prior environments than the standard OG algorithm. The approach relies on a forward model of the behaviour of the plume produced by a vent, which is used to estimate, for each OG cell, the likelihood that a vent in that cell would generate a detection at the location of the vehicle.

In the OG mapping approach we divide the search area into a grid of C cells, each of which can either contain a vent or not. Since the map is two-dimensional, we restrict the agent to move in two dimensions which also simplifies the planning problem. We also allow the agent to move only between adjacent grid cells. Actions are assumed to be deterministic as AUV ground-tracking navigation is quite reliable. Similarly, we assume that the agent can move one cell in one timestep, and there are a fixed number of timesteps in a mission.

The criteria for evaluating the performance of an algorithm is how many vents it is able to find on a mission. To support this criteria we assume the AUV is able to detect whether or not there is a vent at its current location, so it can keep track of the vents it finds. The state of the agent s, which is not fully observable, is therefore made up of:

- c_{AUV}, the cell the AUV is in.
- $\mathbf{U}$, the ocean current vector.
- $\mathbf{v}$, a length-C vector of booleans marking all vents that have already been found.
- $\mathbf{m}$, a similar vector giving the actual vent locations (*true* for cells that contain a vent, *false* for all other cells).

We assume the agent knows the first three exactly, and the OG represents its belief about the true vent locations $\mathbf{m}$. Then the belief state b is composed of:

- c_{AUV}, the cell the AUV is in; $\mathbf{U}$, the current; and $\mathbf{v}$, the list of previously-found vents.
- The occupancy grid $\mathbf{O}$. $\mathbf{O}$ is a length-C vector of values $P(m_c)$, the probability of cell c containing a vent (which will be either zero or one for visited cells). The OG defines a distribution over true vent locations $\mathbf{m}$.

Finally, we introduce an observation model describing the possible observations Z that the agent can receive:

$$z = \begin{cases} l & \text{if a vent is located in cell } c \\ p & \text{if a plume is detected in cell } c \\ n & \text{if nothing is detected} \end{cases}$$

We also make use of the observation function $P(z|b,a)$, giving the probability of observing $z = \{l, p, n\}$ after taking action a from belief state b. These probabilities are given in

[7], where we derive them from the plume model of Jakuba [2].

III. BELIEF CHANGE MAXIMISATION ALGORITHMS

Entropy reduction is commonly used in robotic mapping to produce a map with minimum uncertainty in the most efficient way possible. The idea has been applied to chemical plume tracing, where the aim is to locate a chemical source rather than produce a map, by Vergassola, Villermaux, and Shraiman [3]. They developed an algorithm called *infotaxis* which was the inspiration for our belief change maximisation approaches. Infotaxis attempts to find a single chemical source in a turbulent current flow where gradient ascent is ineffective due to large breaks in the chemical signal. Infotaxis works by maintaining a continuous distribution over the source location and acting to minimise the entropy of this distribution. First a set of points near the agent is selected, then for each point $\mathbf{r}$ in this set the change in entropy expected from moving to that point is evaluated:

$$\begin{aligned}\Delta H(\mathbf{r}) &= P_t(\mathbf{r})\,[-H] \\ &\quad + (1 - P_t(\mathbf{r}))\,[\rho_0(\mathbf{r})\Delta H_0 + \rho_1(\mathbf{r})\Delta H_1 + \ldots] \quad (1)\end{aligned}$$

where $P_t(\mathbf{r})$ is the probability of finding the source at $\mathbf{r}$, and ρ_k is the probability of k chemical signal detections at $\mathbf{r}$, with ΔH_k the corresponding change in entropy.

We now describe how we have implemented infotaxis in our domain. With an occupancy grid, the entropy of each cell is given by

$$H_c(b) = -P(m_c)\log P(m_c) - (1 - P(m_c))\log(1 - P(m_c)) \quad (2)$$

which is written as a function of the belief state b to make clear that the $P(m_c)$ values come from a specific OG. An OG does not represent a probability distribution, as the occupancy of each cell is calculated independently. However, this independence of cell probabilities means we can sum cell entropies to find the total entropy of the OG, $H(b) = \sum_c H_c(b)$ [9]. The expected entropy after action a is

$$\mathrm{E}\,[H(b,a)] = \sum_z P(z|b,a) \sum_c H_c(b') \quad (3)$$

where b' is the OG that results from applying Jakuba's algorithm to the belief state b, action a and observation z, which we denote by $b' = \mathrm{srog}(b,a,z)$. Note we have summed over our observation set $z = \{l,p,n\}$, rather than allowing multiple detections per timestep as in [3]. Vergassola et al. point out that the first term in (1) makes the agent exploit its current map, as the entropy drops to zero if the source is found. While we have an equivalent term in (3) (the summation element where $z = l$), allowing multiple sources means the entropy does not drop to zero when we find a vent, so our implementation of infotaxis has slightly less motivation to actually visit sources. Finally, as in [3], we select the action that minimises (3) (i.e., maximises the reduction in entropy). Results using this algorithm are discussed in Section III-B.

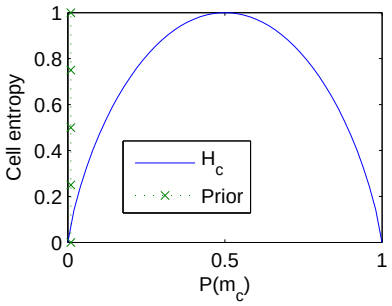

Fig. 2. Plot of entropy of a cell against occupancy probability, with a prior occupancy probability of 0.01 also plotted.

A. $\Sigma\Delta H$ Algorithms

We now present the belief-change-maximisation algorithm we developed, inspired by infotaxis. We do not expect to find many vents in a given search grid, so the prior probability of occupancy of cells in the OG is very low (we use 0.01). This means that when the agent gets a plume detection, which is much more useful to it than no detection, the entropy of affected cells actually increases. Fig. 2 shows a plot of cell entropy together with the low prior to illustrate the problem.

A possible solution to this problem is to initialise the OG with a uniform prior, setting the prior occupancy probability to the maximum entropy value of 0.5. Unfortunately this throws away a lot of information, and the maps produced by the OG algorithm using a uniform prior are inferior to the extent that infotaxis based on these maps is barely better than exhaustive search. Instead our $\Sigma\Delta H$ algorithm[1] fixes this issue by preferring actions that *change* the entropy of cells in the OG the most, regardless of whether the entropy of each cell actually increases or decreases. More precisely, we select the action a that maximises $\mathrm{E}_z\,[\Sigma\Delta H]$, where

$$\Sigma\Delta H(b,a,z) = \sum_c |H_c(b') - H_c(b)| \quad (4)$$

This makes sense because the desirable observations (finding a vent or detecting a plume) result in large changes in $P(m_c)$ values, whereas detecting nothing results in only a small shift in $P(m_c)$ values. We found $\Sigma\Delta H$ to perform well (see Section III-B), but its behaviour appeared myopic, for example it would fail to explore high-probability areas of the grid if it would have to cross its path (consisting of zero-probability cells) to get there (see Fig. 3).

Infotaxis and $\Sigma\Delta H$ only consider the effects of one action, and better plans can generally be formed by anticipating outcomes further into the future. Therefore we developed the $\Sigma\Delta H$-MDP algorithm, which attempts to maximise $\Sigma\Delta H$ values over a discounted infinite horizon. The Bellman optimality equation for action-values Q to maximise $\Sigma\Delta H$

[1]We have nicknamed this family of algorithms "frat-house algorithms" due to the prevalence of Greek letters (Sigma-Delta-Eta).

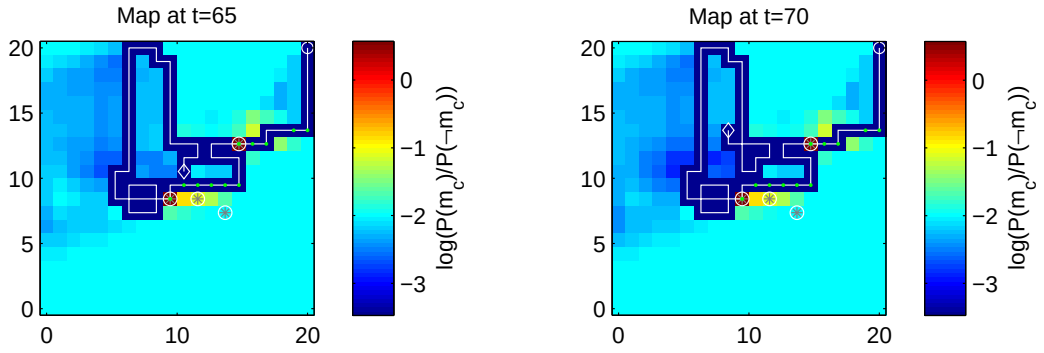

Fig. 3. The agent's path (in white) plotted on top of OG values, showing an example of the myopic behaviour of $\Sigma\Delta H$. The agent's location is shown by a white diamond, and the (unknown to the agent) vent locations are shown by white circles containing grey crosses. In the left-hand plot the agent is very close to an area of high probability where there is a vent, but by the right-hand plot it has turned away from that area because it is 'blocked' by the zero-probability cells it has already explored.

on each step is given by

$$Q(b,a) = \left(\sum_z P(z|b,a)\Sigma\Delta H(b,a,z)\right) + \gamma \sum_z P(z|b,a) \max_{a'} Q(b',a') \quad (5)$$

where γ is a discount factor, $0 \leq \gamma < 1$, that weights rewards lower the further into the future they are. Note that the $\Sigma\Delta H$ algorithm is a special case of this where $\gamma = 0$. Equation 5 cannot be solved iteratively as a set of $|b|$ equations because the belief state b is continuous and cannot be enumerated. It can be solved by forward search through the belief space, but this is exponential in the number of steps of lookahead and therefore practical only for very small lookaheads. Instead we opt to separate the belief space into two components, $b = (\mathbf{O}, c)$, the OG plus the agent's location, and assume that the OG component is fixed. Given a fixed OG, (5) can be collapsed to

$$Q(c,c') = \left(\sum_z P(z|b,c')\Sigma\Delta H(b,c',z)\right) + \gamma \max_{c''} Q(c',c'') \quad (6)$$

where c' represents a cell index adjacent to c (i.e. a possible action), and we have used the starting belief state b to calculate $\mathrm{E}_z[\Sigma\Delta H]$. Equation 6 can be solved efficiently as a normal, discrete-state MDP, with the 'rewards' $\mathrm{E}_z[\Sigma\Delta H]$ being calculated for all grid cells a before running value iteration.

The intuition behind the $\Sigma\Delta H$-MDP algorithm is that we have ignored the effect of observations on the map when the robot moves to a new location, and instead we use the observation function when calculating the 'rewards' $\Sigma\Delta H$ we expect at each location. This allows us to make decisions based on expected observations far away, under the assumption that the mean OG will not change much as we travel to that location, without having to search over an information-state tree with a large branching factor.

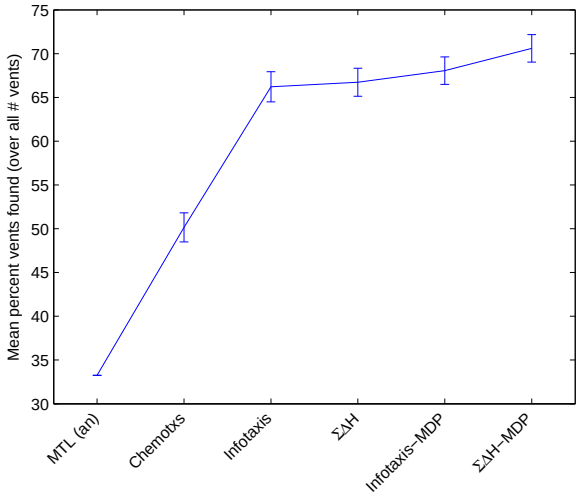

Fig. 4. Results of infotaxis, infotaxis-MDP, $\Sigma\Delta H$, and $\Sigma\Delta H$-MDP. Also shown for comparison are MTL and chemotaxis. All results are means of 600 trials – 150 each with 3, 4, 5, and 6 vents. 95% confidence interval bars are also plotted.

B. Experiments and Results

We used two algorithms to compare our results to: mowing-the-lawn (MTL) and chemotaxis. MTL follows a fixed path where the vehicle moves along parallel tracklines linked by short perpendicular segments. Chemotaxis is a reactive algorithm inspired by the behaviour of moths, where the agent 'surges' up-current when it gets a plume detection, and then moves in a spiral until it gets another detection (see [10]). If the agent attempts to leave the search area, it is directed on a randomly-chosen angle roughly back towards the centre of the grid, and prior to any detection the agent follows an MTL path.

For our experiments, we used a 20×20 grid, and a fixed horizon of 133 time-steps (corresponding to covering $1/3$ of the grid). Equal numbers of trials were conducted with 3, 4, 5 and 6 vents, and these vents were distributed

randomly and uniformly over the search area. The same simulation of current was used for every trial. This current was the same at all points of the grid, but was varied over time by changing the magnitude of the y-component along a sine-wave pattern. The occupancy grid prior was set to $P(m_c) = 0.01$, independent of the number of vents. For each algorithm, 600 trials were performed, and the same set of 600 random seeds was used each time to increase fairness, where the random seed determined both the placement of vents and the dispersion of plume particles from the vents.

The evaluation criteria for our algorithms is how many vents are found during a 133-step mission, and results are shown in Fig. 4. For MTL, instead of running the experiments we display the expected fraction of vents found given infinite trials, i.e. $\frac{133}{400}$, and for $\Sigma\Delta H$-MDP a discount factor of $\gamma = 0.9$ was used in solving the MDP. In the graphs, we show the mean percentage of vents found, where trials are weighted equally regardless of the number of vents in that trial.

Fig. 4 shows that both infotaxis and $\Sigma\Delta H$ are far better at finding vents than chemotaxis or MTL. $\Sigma\Delta H$-MDP improves on $\Sigma\Delta H$, validating our intuition that $\Sigma\Delta H$ would benefit from a longer planning horizon. Further, $\Sigma\Delta H$ improves on infotaxis, and $\Sigma\Delta H$-MDP improves on an MDP version of infotaxis to a statistically significant extent. Experiments with a single vent suggested that infotaxis slightly outperforms $\Sigma\Delta H$ but $\Sigma\Delta H$-MDP is better than both.

IV. CORRECTION USING THE ORIENTEERING PROBLEM

A. MDP Issue and OP Solution

In addition to the explicit assumption that the OG does not change, the $\Sigma\Delta H$-MDP algorithm makes another implicit assumption: that repeated observations from the same cell have the same value as the first observation. In reality this is not the case, as the same plumes will be detectable from any given location. We find that (for areas of the OG unaffected by plume detections) previously-visited cells have $\mathrm{E_z}\left[\Sigma\Delta H\right]$ values about half of those of unvisited cells, a discrepancy which is exaggerated in MDP value functions, as re-visiting a cell with slightly higher reward than surrounding cells will lead to a much higher value. This led us to postulate that a better assumption is that repeated observations have zero value. To encode this in the MDP presented in Section III-A would require a list of previously-visited cells to be added to the state, as the reward for visiting these cells would be zero. Adding this list would increase the state space size from C to $C \cdot 2^C$ possible states, assuming the list was encoded as a boolean vector, meaning $\sim 10^{123}$ states given our experimental settings. While algorithms such as LAO* [11] and RTDP [12] are able to solve very large MDPs, they rely on a heuristic to estimate the distance to the goal. As we define a reward for every cell in the grid, there are no obvious heuristics to estimate how close a given path through the grid is to the globally optimal path. However this problem can be solved by forward search over the action space of the problem, i.e. (6) can be solved recursively, provided we keep track of visited cells during the recursion. Computationally this is a much less demanding problem than forward search in the action-belief space because the occupancy grid does not have to be updated on each step (plus the branching factor is lower as only actions, which are deterministic, need be considered).

This revised problem is in fact identical to the orienteering problem (OP) [6], a variant of the travelling salesman problem where cities are assigned a score, and the agent should choose a subset of cities to visit together with a path between these cities so as to maximise their score, while respecting an upper limit on the distance they can travel. In our OG setting, each cell represents a city, and the distance between all connected cities is the same (as cities are only connected to adjacent cities one grid cell north, east, south and west). The mission duration sets a limit on the number of steps that we wish to maximise our score over.

The OP is still NP-hard, so a fast approximate algorithm was implemented. As the 'cities' are not fully interconnected, many of the heuristics for solving the OP suggested in the literature, which rely on route improvement by swapping two cities [13], [14], could not be easily adopted. Instead we use a Monte-Carlo approach similar to [6], whereby a set of sample paths is generated, and the best path is chosen from these samples. However even for this approach the limited connectivity of the graph causes problems, as we must prevent the path from crossing itself at any point, so we avoid counting the same cell twice in the path value (note that for any path that does cross itself, there is a non-crossing path visiting the same cells plus one extra cell that must be higher-value than the crossing path). This means that we must generate a *self-avoiding walk* through the grid cells, which is itself a complex problem for which most algorithms are exponential in the length of the walk [15]. We generate walks by adding a randomly-chosen action to the walk, provided at least one of the possible actions is not already part of the path, or starting again if the path has reached a 'dead end'. This *inversely restricted sampling* is much more efficient than simple sampling methods [15], but is still exponential in the walk length N. For walks of length $N \leqslant 8$, we use an optimal exhaustive enumeration procedure, as this is faster than the Monte-Carlo algorithm for such short walks.

We use the OP solver to create a variant of the $\Sigma\Delta H$-MDP algorithm: instead of solving an MDP in $\mathrm{E_z}\left[\Sigma\Delta H\right]$ values, we solve an N-step OP where the reward for each step is the $\mathrm{E_z}\left[\Sigma\Delta H\right]$ value of that cell. As the OP solver is exponential in the path length N, we have had to cap N at 30 steps, whereas the agent has 133 steps available at the start of the mission. However the affect of this capping is small given that, as in (6), we use a discount factor of $\gamma = 0.9$. Pseudo-code for the $\Sigma\Delta H$-OP30 algorithm (where a 30-step OP is solved at every timestep) is given in Fig. 5. Note that the limited path length the OP is solved for also means that for $\Sigma\Delta H$-OPN, the algorithm is independent of grid size for all grids larger than $2N \times 2N$ cells (as $\Sigma\Delta H$

```
Procedure choose-action(b, c_curr): cell
  For each cell s
    For each observation z
      calculate P(z|b,s)
      Set b' = srog(b,s,z)
    Set E_ΣΔH(s) = Σ_z P(z|b,s) Σ_c |H_c(b') − H_c(b)|
  Set path=solve-op(c_curr, E_ΣΔH, 30)
  Return path(1)

Procedure solve-op(start,V,num-steps): array
  Set bestValue=-1
  Set bestPath=[]
  Repeat 3000 times
    Set trial={randomly-generated SAW of
               length num-steps starting
               at adjacent_cell(start)}
    If Σ_{t∈trial} V(t)>bestValue
      Set bestValue=Σ_{t∈trial} V(t)
      Set bestPath=trial
  Return bestPath
```

Fig. 5. Pseudo-code for the $\Sigma\Delta H$-OP30 algorithm, where the procedure `choose-action` is applied at each timestep. The OP part of the algorithm tries to maximise the total discounted $\Sigma\Delta H$ value of a path of length 30 steps (for clarity, the discounting by a factor of γ on each step is not shown in the pseudo-code).

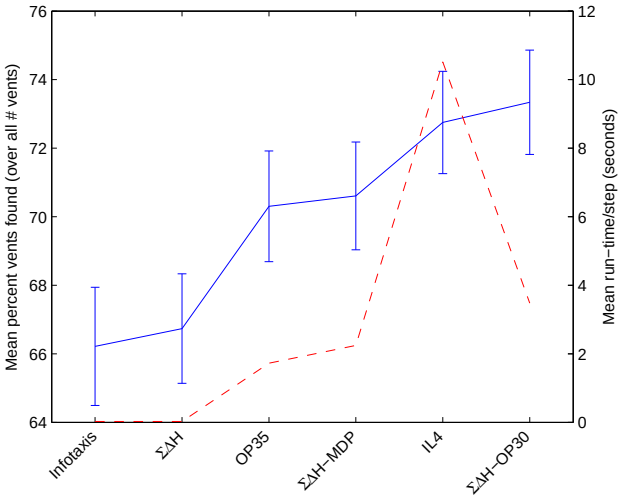

Fig. 6. Results for the $\Sigma\Delta H$-OP30 algorithm compared to OP35, IL4, Infotaxis and the other belief-change-maximisation algorithms. The solid line is the mean percentage of vents found, and the dashed line is the run-time per time-step. All results are means of 600 trials – 150 each with 3,4,5, and 6 vents. 95% confidence bars are also plotted.

values further than N steps from the agent do not have to be calculated).

B. Application to IL Algorithms

As well as improving the performance of our $\Sigma\Delta H$-MDP algorithm, the OP heuristic can be applied in any situation where the reward for revisiting a state is zero, but formulating the problem as an MDP still provides a useful approximation despite the non-Markovian state. In particular it can be applied to the *information-lookahead certainty-equivalent* (IL-CE) algorithm which we have previously developed to solve the hydrothermal vent prospecting problem [7]. In IL-CE we use forward search in the action-belief state space for several steps (the IL part, essentially solving an online POMDP), and then assume the OG is fixed and solve an MDP to provide the leaf-node values (the CE part). IL-CE requires the definition of a reward function $R(b,c') = R_{vent}P(m_{c'})$, where R_{vent} is a fixed reward the agent receives for discovering a new vent. Results showed that using an MDP to provide leaf-node values actually harmed performance compared to simply truncating the forward search (i.e. using just IL), due to the interaction between the forward lookahead and the MDP values. The problem we encountered was that MDP values for some cells could become very large, as the agent can re-visit a high reward cell repeatedly. This meant that the optimal plan for the IL lookahead (which was performed for four steps) was to avoid these high-reward cells, so that they could be visited repeatedly once the agent entered the MDP phase. The resultant behaviour was that the agent never visited high-value cells because it re-planned on every step, always planning to visit these high value cells in four steps time.

Applying the OP heuristic in place of the MDP is an obvious way to fix this difficulty, as then re-visiting cells would have no value in either the lookahead phase or the leaf-node heuristic phase of planning. The only issue is that both the IL and OP algorithms are exponential in the number of steps, which severely limits the OP path length we can use despite the fact that OP is much faster than IL. For example, given that there is a branching factor of nine for the action-belief lookahead (three non-backtracking actions and three observations), for four steps of IL lookahead the OP must be solved about 6500 times on every time-step of the simulation.

C. Results and Discussion

Fig. 6 shows comparative results for $\Sigma\Delta H$-OP30, and OP35 using the same experiment conditions described in Section III-B. It is clear that $\Sigma\Delta H$-OP30 is an improvement on $\Sigma\Delta H$-MDP for only a small increase in run-time, with the results being better at a significance level of 0.01 in a one-tailed test. $\Sigma\Delta H$-OP30 is also better than the previous state-of-the-art, IL4 (although not by a statistically significant margin), and runs much faster.

Our results for applying the OP correction to IL are promising, but less clear-cut than for the $\Sigma\Delta H$ case, as Fig. 7(a) shows. For IL2 (two steps of information lookahead), adding an OP correction invariably improved the performance, but increasing the OP path length above two did not result in any further systematic improvement. Fig. 7(a) shows IL2-OP8 and IL2-OP14, which are the best and worst of all the OP corrections respectively (experimental results for IL2-OP$\{2,4,6,10,12\}$ are omitted in the interests of clarity). With four steps of IL, adding an OP6 correction actually results in worse performance, but early results show this is reversed for IL4-OP10.

We hypothesise that the OP correction interacts in a complex manner with the exploration/exploitation tradeoff inherent in the IL algorithm. Note that the IL algorithm would be optimal if it could be run with a large enough lookahead, whereas the OP correction is greedy—it will try

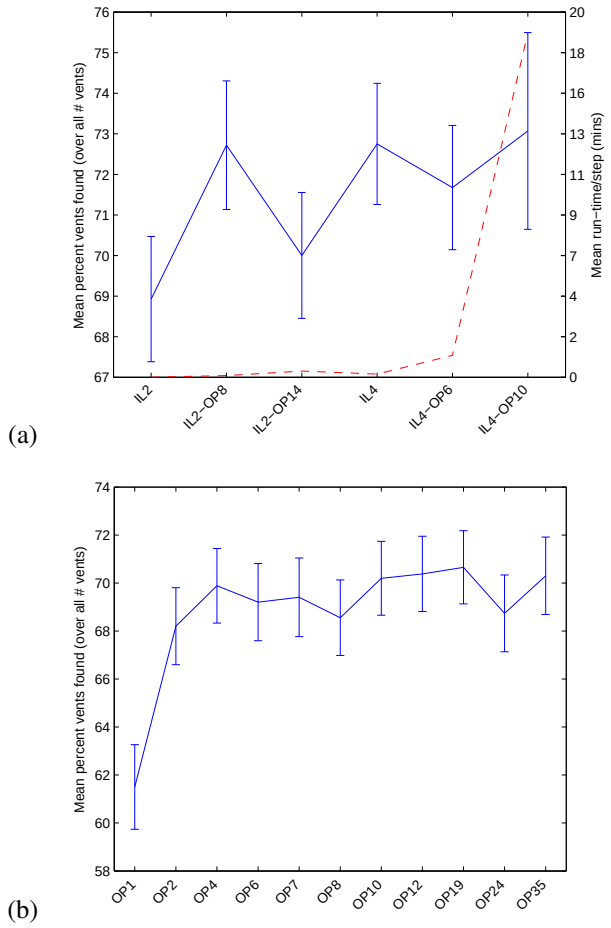

Fig. 7. Effects of the OP correction. (a): Effects of the OP correction on IL2 and IL4, with run-time per time-step shown as a dashed red line. Note the 95% confidence intervals are larger for IL4-OP10 because they are over 240 trials rather than 600 for all other algorithms, due to computational constraints. (b): Performance of the pure-OP algorithm against OP-path-length (note x-axis is not to scale in terms of path-length).

to visit all cells in the vicinity of a vent with $P(m_c)$ even slightly higher than the prior instead of choosing to abandon that area and search for more vents. Our intuition is that having a larger OP correction partly cancels out the drive to perform exploratory actions in IL, causing the agent to spend too much time near vents it has already localised.

Fig. 7(b) shows the effect of increasing the OP path for a pure-OP algorithm where the values are $R(b, c') = R_{vent}P(m_{c'})$, which defines the expected immediate reward if R_{vent} is a fixed reward the agent receives for discovering a new vent. This shows that a simple gradient-ascent algorithm (OP1) is far from optimal in this environment, but also that increasing the OP path does not consistently improve performance.

V. RELATED WORK

Occupancy grids [16] are a common approach in mobile robotic planning applications and entropy measures are frequently used to decide how to explore in these representations. For example, Bourgault et al. [4] and Stachniss et al. [17] both choose actions that maximise the reduction in both the robot pose entropy and the occupancy grid entropy for a robot building a map. Bourgault et al. require identifiable landmarks for mapping, while Stachniss et al. relax this. In both cases, the occupancy grid is updated using data from a laser rangefinder. These approaches differ from our domain in that the priors for the grid cells in their mapping problem tend to be 0.5. This means that any observation about a cell reduces the entropy about that cell, whereas in our case an observation of a plume will actually increase the entropy of some grid cells. A similar approach was taken in an underwater mapping domain by Low et al. [18]. Again their approach is not directly relevant to our problem as they are map building rather than looking for particular distinguished cells.

In terms of previous work in chemical plume source finding, we have already mentioned the work of Vergassola et al. There are other related approaches such as [10], which uses purely reactive approaches to follow a plume to its source and allow the current carrying the plume to be turbulent, but they all make the same assumption Vergassola does of a single source.

Finally, recent work has leveraged the submodular property of sensor placements, that each additional observation is worth less than the preceding one, to develop efficient algorithms for planning autonomous survey paths [19]. However, this work relies on sensors having a localised observation field, so that distant measurements are independent, which is not the case in our domain where distant vents can be detected.

VI. CONCLUSIONS

We have described a novel belief-change-maximisation algorithm intended for use as an on-board planner on a robot searching for landmarks of interest in a low-prior occupancy grid environment. This $\Sigma\Delta H$-MDP algorithm chooses actions that result in the largest mean *change* in entropy of OG cells. It is able to plan ahead over a discounted infinite horizon by ignoring belief-state changes for movement actions but utilising expected belief-state changes for making observations at distant locations. Additionally, we have introduced the use of an OP solver to replace the MDP and improve the valuation of repeated visits to grid cells. We evaluated our algorithms on a simulation of hydrothermal vent prospecting using an AUV, and found that $\Sigma\Delta H$-OP provides performance at least equivalent to information-lookahead in the action-belief space, which is the best method previously tried in this environment, but with considerably reduced computation time.

We also applied the OP solver as a correction to information lookahead, where it shows promise in improving the algorithm's performance. However there appear to be complex interactions here, possibly due to the exploration/exploitation trade-off that must be made in this domain, and these are worthy of further research. Also this algorithm is of limited practical relevance due to the lengthy computation time which makes it impractical for use on an actual AUV.

In addition to further work on investigating the effect of the OP correction on information-lookahead, our next goals include implementing a more efficient OP solver. While we want an OP algorithm that scales much better with the path length N, a significant constraint is that we need the algorithm to run very quickly. For example, our current code takes a lengthy 2.5 hours to execute an IL4-OP6 trial, but to achieve even this performance the OP solver has to run in less than 0.01 seconds (due to the large number of belief states generated by looking ahead four steps). More sophisticated OP algorithms are given in [20], but given our performance requirements, a better approach may be an improved algorithm for self-avoiding walks.

REFERENCES

[1] R. Dearden, Z. Saigol, J. Wyatt, and B. Murton, "Planning for AUVs: Dealing with a continuous partially-observable environment," in *Workshop on Planning and Plan Execution for Real-World Systems, ICAPS-07*, 2007. [Online]. Available: http://eprints.bham.ac.uk/236/

[2] M. Jakuba, "Stochastic mapping for chemical plume source localization with application to autonomous hydrothermal vent discovery," Ph.D. dissertation, MIT and WHOI Joint Program, 2007. [Online]. Available: http://robotics.me.jhu.edu/~jakuba/research/jakuba_final.pdf

[3] M. Vergassola, E. Villermaux, and B. I. Shraiman, "'Infotaxis' as a strategy for searching without gradients," *Nature*, vol. 445, no. 7126, pp. 406–409, 2007. [Online]. Available: http://dx.doi.org/10.1038/nature05464

[4] F. Bourgault, A. Makarenko, S. Williams, B. Grocholski, and F. Durrant-Whyte, "Information based adaptive robotic exploration," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-02)*, 2002, pp. 540–545. [Online]. Available: http://dx.doi.org/10.1109/IRDS.2002.1041446

[5] J. Farrell, S. Pang, W. Li, and R. Arrieta, "Chemical plume tracing experimental results with a REMUS AUV," *Oceans Conference Record (IEEE)*, vol. 2, pp. 962–968, 2003.

[6] T. Tsiligirides, "Heuristic methods applied to orienteering," *Journal of the Operational Research Society*, vol. 35, no. 9, pp. 797–809, 1984. [Online]. Available: http://www.jstor.org/stable/2582629

[7] Z. A. Saigol, R. W. Dearden, J. L. Wyatt, and B. J. Murton, "Information-lookahead planning for AUV mapping," in *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*, 2009. [Online]. Available: http://ijcai.org/papers09/Papers/IJCAI09-304.pdf

[8] M. C. Martin and H. Moravec, "Robot evidence grids," CMU Robotics Institute, Tech. Rep. CMU-RI-TR-96-06, 1996. [Online]. Available: http://www.ri.cmu.edu/publication_view.html?pub_id=401

[9] R. Rocha, J. Dias, and A. Carvalho, "Cooperative multi-robot systems:: A study of vision-based 3-d mapping using information theory," *Robotics and Autonomous Systems*, vol. 53, no. 3-4, pp. 282–311, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/B6V16-4HG69TT-1/2/e15de87abc74a84e44dbe72206f2d4b6

[10] R. A. Russell, A. Bab-Hadiashar, R. L. Shepherd, and G. G. Wallace, "A comparison of reactive robot chemotaxis algorithms," *Robotics and Autonomous Systems*, vol. 45, no. 2, pp. 83–97, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/B6V16-49H721Y-1/2/6d32dab522401992311c75d1ea82bac0

[11] E. A. Hansen and S. Zilberstein, "Lao*: A heuristic search algorithm that finds solutions with loops," *Artificial Intelligence*, vol. 129, no. 1-2, pp. 35–62, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYF-44MS5W2-3/2/06595190f54686cee2933c257253e957

[12] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 81–138, 1995, q-learning algorithm;. [Online]. Available: http://dx.doi.org/10.1016/0004-3702(94)00011-O

[13] B. Golden, L. Levy, and R. Vohra, "The orienteering problem," *Naval Research Logistics*, vol. 34, no. 3, pp. 307–318, 1987. [Online]. Available: http://dx.doi.org/10.1002/1520-6750(198706)34:3⟨307::AID-NAV3220340302⟩3.0.CO;2-D

[14] I.-M. Chao, B. Golden, and E. Wasil, "A fast and effective heuristic for the orienteering problem," *European Journal of Operational Research*, vol. 88, pp. 475–489, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/B6VCT-3VW1D5N-1D/2/f6797a910c7d46471b7d3bd58b28e8fe

[15] A. D. Sokal, *Monte Carlo and Molecular Dynamics Simulations in Polymer Science*. Oxford University Press, New York, 1995, ch. Monte Carlo Methods for the Self-Avoiding Walk. [Online]. Available: http://arxiv.org/abs/hep-lat/9405016v1

[16] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989. [Online]. Available: http://dx.doi.org/10.1109/2.30720

[17] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *Proc. of Robotics, Science and Systems*, 2005, pp. 65–72.

[18] K. H. Low, J. M. Dolany, and P. Khosla, "Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing," in *Proc. of the Nineteenth International Conference on Automated Planning and Scheduling*, 2009, pp. 233–240. [Online]. Available: http://www.aaai.org/ocs/index.php/ICAPS/ICAPS09/paper/viewFile/699/1117

[19] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *Proc. of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07)*, 2007, pp. 1650–1654. [Online]. Available: http://www.aaai.org/Papers/AAAI/2007/AAAI07-265.pdf

[20] A. Blum, S. Chawla, D. Karger, T. Lane, A. Meyerson, and M. Minkoff, "Approximation algorithms for orienteering and discounted-reward TSP," in *Proc. 44th IEEE Symposium on Foundations of Computer Science (FOCS-03)*, 2003, pp. 46–55. [Online]. Available: http://dx.doi.org/10.1109/SFCS.2003.1238180