# Explore vs. Exploit: Task Allocation for Multi-robot Foraging

Christopher Baldassano† and Naomi Ehrich Leonard‡
†Electrical Engineering and ‡Mechanical and Aerospace Engineering
Princeton University, Princeton, NJ 08544, USA

*Abstract*— **This paper describes two measures of performance that can be used to allocate tasks in a multi-robot foraging problem. These heuristics can be used by a human supervisor or an automated control algorithm to adjust the number of robots exploring the environment versus the number of robots greedily harvesting based on current knowledge. A numerical simulation study is presented that offers preliminary support for the usefulness of these heuristics.**

## I. INTRODUCTION

A decision-making framework is proposed for allocating tasks to a homogeneous group of robots in an unexplored environment. It is assumed that every robot can be assigned to one of two tasks: *exploration* (large-scale information gathering) or *exploitation* (focused collection of resources). This framework can be applied to any harvesting task in which the maximum resource collection rate varies both spatially and over time; for example, a model problem is positioning mobile solar panels under changing lighting conditions.

In any problem of this type, there is a nontrivial question of how to best allocate the robots between exploration and exploitation. Assigning many robots to be explorers will increase the number of areas of interest identified, but will decrease the number of robots able to exploit these areas and could lead to redundancy in mapping. Assigning many robots to be exploiters will increase the collection rate when the field is well-known, but could be ineffective if the robots are not able to position themselves intelligently in their environment.

Optimizing this allocation requires a comprehensive understanding of the characteristics of the robots and the field being explored, and it is not clear whether the robots can make the decision without human aid. If it is necessary to involve a human supervisor, the division of labor between the robots and the human for this allocation task should take into account the respective strengths of the humans and robots. In this paper two heuristics are introduced, which can be calculated by the robots in order to aid a human or automated supervisor in selecting an allocation. A preliminary simulation study is presented to help assess the value of these heuristics and to draw insights on the allocation problem.

This research was supported by AFOSR grant FA9550-07-1-0-0528.

## II. LITERATURE REVIEW

Multi-Robot Task Allocation (MRTA) has been investigated in a variety of settings, but there are few general strategies since most work has focused on architectures for specific systems, such as UAVs [1]. Other work on task allocation for exploration has assumed there are a discrete number of known targets to be investigated [2], which is not applicable for a continuous, unknown field. There has been some work to develop a formal framework for studying MRTA problems [3], with empirical investigation into selecting dynamic task allocation strategies [4].

Optimal policies for choosing between exploration and exploitation have been presented for the limited class of problems in which a single agent chooses among multiple options with stationary pay-off structures and agents discount future rewards exponentially. In this case, the optimal policy can be determined by computing the Gittins indices for each option, although this computation may be intractable [5], [6]. Time-dependent reward structures have been previously investigated in biological settings, in the field of optimal foraging theory [6], [7], [8]. Models and optimal solutions have been developed for very simple situations, such as a single bird exploring two food sources before choosing one of them [8]. There is, however, no known general strategy [6]. A new study examines the dynamics of exploration versus exploitation by teams of sensor-enabled robots mapping potential fields that are possibly time-varying [9]. Decisions between search strategies are made using a mutual information-like measure with the aim of finding the optimally aggressive exploration strategy.

Possible assessments of performance used by animals harvesting discrete units of resources have been proposed, such as time since last encounter of an item and rate of encounter of items [7]. Measuring the rate of encounter can be accomplished using Green's assessment rule, and this rule has been applied in modified form to human task switching [10]. The appropriate heuristics and the degree to which the allocation decision can be performed without human supervision are still open questions.

In general, systems that actively manage sensors based on the current (estimated) state of the environment fall into the category of *adaptive sensing*. Recently,

these problems have been formalized as corresponding to a Partially Observable Markov Decision Process (POMDP), for which optimal actions can be chosen through Q-function maximization [11]. One proposed method of approximating the Q-function is to use domain knowledge to generate a heuristic ranking of Q-values for possible actions [11]. The harvesting task is a generalization of adaptive sensing, since the objective is not just to measure the environment but also to devote robots to acting on that information. Although this difference and the massive state space of our problem preclude us from using the adaptive sensing POMDP framework directly, the idea of ranking current decisions based on heuristics is still applicable.

Given an allocation, explorers and exploiters will need to follow an algorithm for autonomously performing their tasks. Control systems for multi-robot exploration have been investigated by a number of researchers [12], [13], [9], [14]. Simple control algorithms are used here as proxies for a more sophisticated approach.

## III. PROBLEM DEFINITION

The approach described in this paper is applicable to allocation problems with the following features:

- A fixed number $N$ of homogeneous mobile robots in a spatial domain, each of which can collect resources, make local field measurements, and move with maximum speed $v_{exploit}$ when harvesting (exploiting) and $v_{explore}$ when exploring.
- A scalar field $R(\mathbf{r}, t)$ describing the maximum rate of resource collection at each point in time and space, with known mean $\bar{R}$ and covariance

$$B(\mathbf{r}, t, \mathbf{r}', t') = \sigma_0 \exp^{-\frac{|\mathbf{r}-\mathbf{r}'|^2}{\sigma^2} - \frac{|t-t'|^2}{\tau^2}} \quad (1)$$

  where $\sigma$ and $\tau$ define the time and length scales of the covariance.

Note that the field may be negative at some points; thus $R(\mathbf{r}, t)$ is more correctly interpreted as the net resource collection rate (including costs of running the robot), which can be negative.

## IV. HEURISTIC APPROACH

Given the intractability of exact solutions to the allocation problem, a simpler approach is to develop heuristics to allow for approximate comparisons between alternative allocations. The robots or a human supervisor can then monitor the value of these heuristics and modify the allocation appropriately. The following sections describe two heuristics that can be used simultaneously to evaluate the performance of the robots. The first focuses on the actions of the explorers, while the second is based on the actions of the exploiters. Let $m(t)$ be the number of exploiters at time step $t$ and $n(t) = N - m(t)$ the number of explorers.

### A. Explorer Heuristic

This heuristic makes use of the information metric defined in [12]. Given a sequence of $P$ measurements $R_k$ at points $(\mathbf{r}_k, t_k)$, and the field covariance $B(\mathbf{r}, t, \mathbf{r}', t')$, the a posteriori error is

$$\begin{aligned} A(\mathbf{r}, t, \mathbf{r}', t') &= B(\mathbf{r}, t, \mathbf{r}', t') - \sum_{k,l=1}^{P} B(\mathbf{r}, t, \mathbf{r}_k, t_k) \\ &\quad * (C^{-1})_{kl} * B(\mathbf{r}_l, t_l, \mathbf{r}', t') \end{aligned} \quad (2)$$

where $C^{-1}$ is the inverse of the covariance for the data points, the elements of $C$ given by

$$(C)_{kl} = \tilde{n}\delta_{kl} + B(\mathbf{r}_k, t_k, \mathbf{r}_l, t_l) \quad (3)$$

and $n$ the measurement noise. The entropic information of the area $\mathcal{A}$ of domain $\mathcal{D}$ at time step $t$ is defined as

$$I(t) = -\log\left(\frac{1}{\sigma_0 \mathcal{A}} \int d\mathbf{r} A(\mathbf{r}, t, \mathbf{r}, t)\right). \quad (4)$$

The information $I(t)$ will decrease with time if no new measurements are made (since the field has a time-scale $\tau$), so it is not appropriate to measure the information gained by the explorers as $I(t) - I(t-1)$ since even keeping $I(t)$ constant requires continued exploration. Instead, define the *incremental information* as

$$\widetilde{I}(t) = I(t)_{Data\,[1:t]} - I(t)_{Data\,[1:(t-1)]} \quad (5)$$

where $I(t)_{Data\,[1:T]}$ refers to the information at $t$ given that measurements are taken over the interval of time steps from 1 to $T$. Incremental information is the decrease in the current information $I(t)$ that would occur if the measurements during the previous time step had not been made. This ensures that $\widetilde{I}(t)$ quantifies the informational value of the last round of measurements. This is similar to the concept of *information inflow* in [9].

The informational heuristic is defined as

$$H_{explore}(t) = \widetilde{I}(t) * m(t). \quad (6)$$

This heuristic defines the value of a current allocation based on the usefulness of the information collected, multiplying the incremental information by the number of robots that can act on that information. Information has no value in and of itself, and therefore its usefulness is directly dependent on the number of exploiters that could potentially be helped. Note that this information can help all exploiters over the next (approximately) $\tau$ time steps, so the current number of exploiters is only an estimate of the average number of exploiters helped per time step.

### B. Exploiter Heuristic

Since the current rate of resource collection will be highly volatile as a function of $t$ even under constant allocation (due to robot motion and changes in the field), a more stable estimate is the expected resource collection over the next $\tau$ time steps. Predicting signif-

icantly farther ahead than $\tau$ would be uninformative, since the future values of the field can only be weakly predicted. The estimated rate of resource collection over the next $\tau$ time steps for a single robot, given the current location $\mathbf{r_i}$ and destination $\mathbf{d_i}$ of the robot, is

$$F_i(\mathbf{r_i}, \mathbf{d_i}, t) = \frac{1}{\tau}(\bar{R} * \frac{||\mathbf{r_i} - \mathbf{d_i}||}{v_{exploit}} + \hat{R}(\mathbf{d_i}, t) * (\tau - \frac{||\mathbf{r_i} - \mathbf{d_i}||}{v_{exploit}})) \quad (7)$$

where $\hat{R}(\mathbf{r}, t)$ is the estimated rate of resource collection at point $\mathbf{r}$ and current time $t$ computed using

$$\hat{R}(\mathbf{r}, t) = \bar{R} + \sum_{k=0}^{P} \zeta_k(\mathbf{r}, t) * (R_k - \bar{R}). \quad (8)$$

As in [12] the optimal coefficients $\zeta_k(\mathbf{r}, t)$ that minimize the mean square error with the actual field $R(\mathbf{r}, t)$ are

$$\zeta_k(\mathbf{r}, t) = \sum_{l=1}^{P} B(\mathbf{r}, t, \mathbf{r}_l, t_l) * (C^{-1})_{kl}. \quad (9)$$

The function $F_i$ makes the assumption that resources are collected at the mean rate en route to the robot's current destination, and that resources will be collected at rate $\hat{R}(\mathbf{d_i}, t)$ in the time between the robot's arrival time and $\tau$ (where the robot's arrival time is estimated by assuming a straight path at maximum speed).

The exploiter heuristic $H_{exploit}$ is defined as

$$H_{exploit} = \sum_{i=1}^{m} F_i(\mathbf{r_i}, \mathbf{d_i}, t). \quad (10)$$

Consider the following simplifying assumptions: each explorer $i$ makes exactly one measurement $N_i$ every $\tau$ time steps (simultaneously); $N_1, \ldots, N_n$ are independent, identically distributed random variables with normal distribution (unit mean and variance); the $m$ exploiters can move instantly, and can all occupy the same point without interfering with one another; the rate of resource collection is constant over every $\tau$ time steps after the measurements are made. In this condition, all of the exploiters will harvest at the best location discovered, so the expected rate of resource collection after all the measurements are made is

$$H_{exploit} = E[\max\{N_1, N_2, ...N_n\}] * m. \quad (11)$$

The expectation can be calculated:

$$\begin{aligned} &P[\max\{N_1, N_2, ...N_n\} \in dk] \\ &= \frac{d}{dk} P[\max\{N_1, N_2, ...N_n\} < k] \\ &= \frac{d}{dk} P[N_1 < k \,\&\, N_2 < k \,\&\, ... \,\&\, N_n < k] \\ &= \frac{d}{dk} (\Phi(k))^n \\ &= n(\Phi(k))^{n-1} N(k) \end{aligned} \quad (12)$$

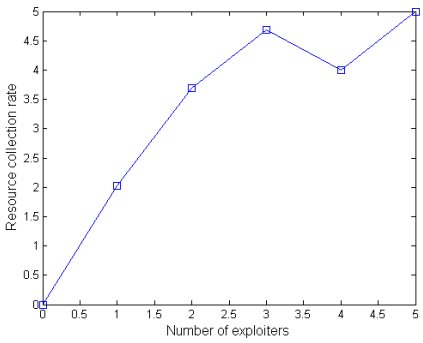


Fig. 1. A theoretical approximation of $H_{exploit}$ as a function of the number of exploiter robots (5 robots total). The decrease in the expected maximum of the explorer measurements is offset by the number of exploiters able to harvest at the maximum.

where $\Phi(k)$ is the cdf for a normal distribution. The heuristic is therefore

$$H_{exploit} = \left[\int_{-\infty}^{\infty} dk * k * n(\Phi(k))^{n-1} N(k)\right] * m. \quad (13)$$

When $n = 0$, then expected rate of resource collection by the exploiters is simply $m$ since the mean of the field is 1. The expected collection rate is graphed in Figure 1 as a function of $m$, for $n + m = 5$.

## V. SIMULATION DESIGN

In order to run a full simulation of a task allocation scenario, automatic planning algorithms were implemented to direct the motion of the individual exploiters and explorers in a planar domain $\mathcal{D}$. The same estimation technique is used, as for the exploiter heuristic above, to locally direct the exploiters:

$$\arg\max_{\mathbf{d_i} \in \mathcal{D}} \left\{ \bar{R} * \frac{||\mathbf{r_i} - \mathbf{d_i}||}{v_{exploit}} + \hat{R}(\mathbf{d_i}, t) * (\tau - \frac{||\mathbf{r_i} - \mathbf{d_i}||}{v_{exploit}}) \right\} \quad (14)$$

A similar formula was used to choose the explorer destinations, treating explorers as consumers of error:

$$\arg \max_{\mathbf{d_i} \in \mathcal{D}: ||\mathbf{r_i} - \mathbf{d_i}|| > v_{explore}} A(\mathbf{d_i}, t) * (\tau_A - \frac{||\mathbf{r_i} - \mathbf{d_i}||}{v_{explore}}) \quad (15)$$

where parameter $\tau_A$ defines an effective maximum search radius. The condition that explorers must set destinations at least $v_{explore}$ away forces them to always move at their maximum speed; the search area for explorer destinations is limited to an annulus. It is also desirable that the explorers not set destinations near the edges of the domain, since measurements made there provide less entropic information (as the circle of radius $\sigma$ is partially outside the domain) and could only find spatially small peaks (which would only fit one or two exploiters at best). Therefore, explorers do not consider destinations within a distance $\sigma$ of the domain limits.

For both algorithms, the destinations are computed sequentially and no robot is allowed to choose a des-

| Constant Name | Symbol | Value |
|---|---|---|
| Domain Side Length | $l$ | 50 |
| Number of Robots | $N$ | 5 |
| Robot Radius | $r$ | 3 |
| Exploiter Speed | $v_{exploit}$ | 5 |
| Explorer Speed | $v_{explore}$ | 6 |
| Measurement Noise | $\tilde{n}$ | .1 |
| Covariance Scaling | $\sigma_0$ | 1 |
| Field Mean | $\bar{R}$ | 1 |
| Field Length Scale | $\sigma$ | 10 |
| Field Time Scale | $\tau$ | 10 |

TABLE I
SIMULATION PARAMETERS

tination within radius $r$ of a destination chosen by another robot. Although collisions are not enforced in the simulation, forcing the destinations to be non-overlapping allows the simulation to incorporate some element of inter-robot interference. Note that this simple sequential algorithm can cause inefficiencies due to the order in which robots choose destinations; a more advanced scheme would allow robots to trade destinations if it would be to their mutual benefit.

Table I summarizes the constants used in the simulation. The domain is a square and $\tau_A = \tau$.

## VI. SIMULATION RESULTS

### A. Control Algorithms

Although not a central focus of the experiment, the destination-setting algorithms exhibited very reasonable behavior in this study. Sample destination decisions for the exploiters and explorers are shown in Fig. 2 and 3 respectively. At no point during any runs of the simulation did the algorithms fail to find destinations matching the constraint conditions described.

The paths of the robots generally appear chaotic, but the special case of exactly one explorer exhibits an interesting emergent behavior. Whenever the four exploiters were clumped together and near-stationary, the explorer would execute ellipses bounded by the exploiters' positions and the limits of its search space (Fig. 4). This connects to previous work in collaborative sensing, in which explorer performance has been explicitly optimized over a family of ellipses [12]; the appearance of ellipses in the present study suggests that elliptical paths may be roughly equivalent to greedy error reduction.

### B. Heuristic Performance

A plot of the exploiter heuristic during a typical simulation is shown in Fig. 5. The heuristic is observed to accomplish the goal of removing noise that could confuse a (human or automated) supervisor. E.g., at time step 139, two of the three exploiters are crossing a resource-poor area to reach a high peak. The resource collection rate drops sharply, but there is no need for more exploration since the exploiters already have a good plan for the next $\tau$ time steps; indeed the heuristic does not decrease so significantly. The exploiter heuristic is much smoother overall than the resource collection rate; the coefficient of variation (standard deviation divided by mean) is .282 for the heuristic vs. .379 for the collection rate (averaged over six 200-time-step trials, each with a unique and constant allocation).

Note that the exploiter heuristic is *not* simply a smoothed or moving-average version of the resource collection rate. Any smoothing filter would necessarily introduce lag, making the system slow to respond to the sharp collection rate increases and decreases that are typical when finding and losing peaks in the resource field. As seen in Fig. 5, the exploiter heuristic tracks these sharp changes nearly exactly, while rejecting noise due to temporary conditions. No function of the collection rate that does not incorporate future predictions could replicate this intelligent behavior.

Fig. 6 is a plot of the explorer heuristic during a simulation. The heuristic tends to oscillate at high frequency due to inefficiency in the explorer paths (resulting in frequent drops in the incremental information) but the average value of the heuristic is relatively stable. The heuristic would be passed through a moving-average

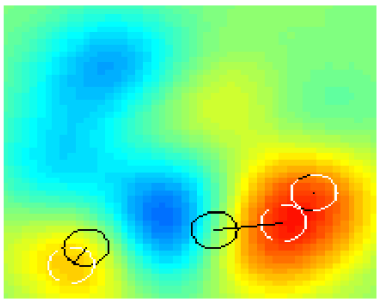

Fig. 2. A sample exploiter destination decision at a specific time. The exploiters (black) and destinations (white) are overlaid on the estimated field $\hat{R}$. Redder regions correspond to higher values of $\hat{R}$. One of the three exploiters is coincident with its destination.

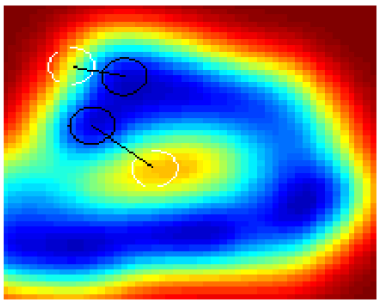

Fig. 3. A sample explorer destination decision at a specific time. The explorers (black) and destinations (white) are overlaid on the error map $A$. Redder regions correspond to larger error.

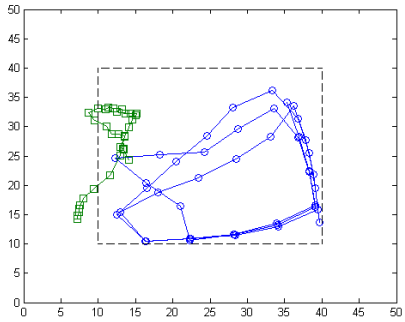

Fig. 4. Paths of robots from time step 31 to 74, with the number of explorers held constant at 1. The mean position of the 4 exploiters (squares in upper-left) remained approximately constant tracking a peak, and the lone explorer (circles in bottom-right) executed ellipses within its planning boundary (dashed line).

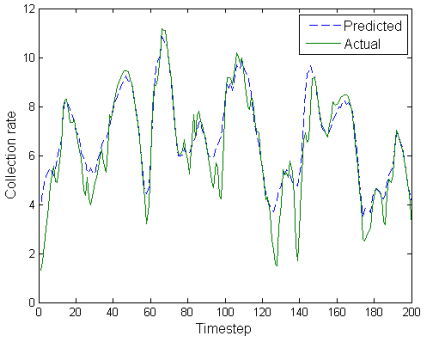


Fig. 5. The exploiter heuristic and actual resource collection rate over 200 time steps in a simulation with 3 exploiters and 2 explorers.

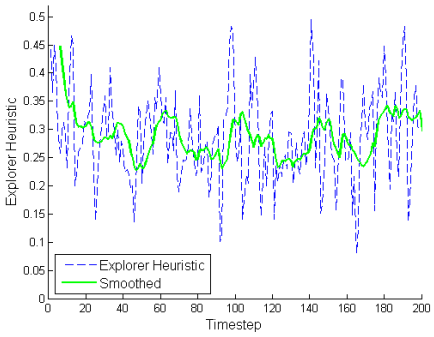


Fig. 6. The explorer heuristic and the result of smoothing (moving-average over $\tau$ time steps) over 200 time steps in a simulation with 3 exploiters and 2 explorers.

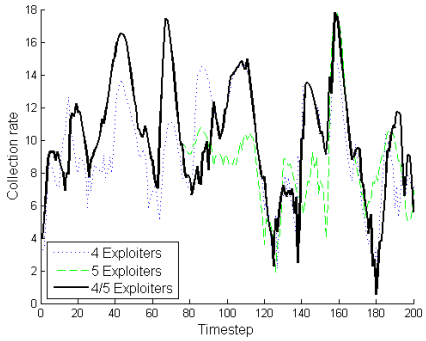


Fig. 7. A comparison of dynamic 4/5 exploiter allocation and static 4 or 5 exploiter allocation. The allocation was manually switched from 5 exploiters to 4 at t = 76 and switched back at t = 153.

filter (shown in the figure) before being reported to the supervisor. Although this does introduce lag (as discussed above) this heuristic is not designed to capture timing information. With a more sophisticated control algorithm, this heuristic would ideally stay nearly constant, with explorers on well-defined paths to reduce error in predictable amounts. The heuristic could even be used to evaluate proposed exploration algorithms; those with high, constant values of incremental information are most effective.

### C. Comparing Allocations

Six 200-time-step trials were run, each with a different (constant) allocation for $N = 5$ robots. For static allocations, it was most rewarding to do only local tracking of peaks. It may be possible to find a set of parameters for which the all-exploiter case does not collect the most resources overall, but it is likely (in agreement with the rough estimation in Fig. 1) that the best static allocation will always be heavily biased in favor of the exploiters. The all-exploiter allocation is not optimal, however, if dynamic allocation is permitted. As demonstrated in Fig. 7, the average resource collection over all 200 time steps can be higher for a dynamic allocation than for *any* static allocation; the total resources collected for the 4 exploiter, 5 exploiter, and dynamic 4/5 exploiter allocations were 1859.3, 1919.3, and 2016.1 respectively. (The speeds in this experiment were adjusted to $v_{explore} = v_{exploit} = 10$ so that the robots could collect and use information more efficiently and better illustrate the significant difference between static and dynamic allocation. Note that, even with these parameter settings, the best static allocation is still the all-exploiter allocation.)

These results suggest that a (human or automated) supervisor determining the best allocation should do so based on *current conditions*. That is, in order to make intelligent dynamic decisions, the supervisor must be provided with information about the current performance of the robots. To make a change in the allocation, the supervisor could consider the current values of our heuristics. There are two immediate questions:

1) *How can a supervisor directly compare the two heuristics, since they have different units?* A scaling factor $\lambda$ must be defined so that $\lambda * H_{explore}$ has units of resources/time step, giving meaning to the relative values of the heuristics. This value will most likely have to be determined experimentally, and could be adjusted to favor exploring or exploiting. It may appear that

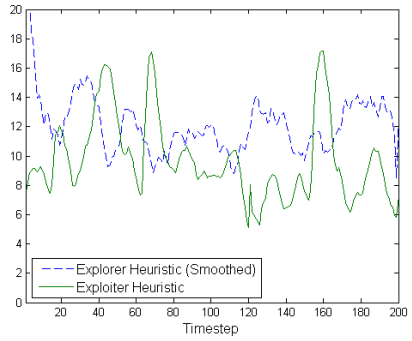


Fig. 8. Heuristic values for the 5 exploiter allocation during the experiment illustrated in Fig. 7.

this has simply brought us back to where we started, since we are forced to choose a trade-off parameter between exploring and exploiting. This single parameter, however, defines the relative values of the heuristics at all future times, and is therefore much simpler than attempting to re-evaluate the tradeoff at every time step. For Fig. 8, $\lambda = 32.6$. This value was found to cause the magnitudes of the heuristics to be equal when the exploiters were collecting about 1.5 resources/time step each, i.e., together about 7.5 resources/time step (a relatively poor collection rate).

2) *How should a supervisor make an allocation decision?*

- The simplest strategy is to maximize $max(H_{exploit}, \lambda * H_{explore})$; when the exploiters are performing well, this is equivalent to maximizing $H_{exploit}$, and when the exploiters are performing poorly, this is equivalent to maximizing $H_{explore}$.
- A more complicated strategy would be to maximize a linear combination $H_{exploit} + \alpha * \lambda * H_{explore}$ (for some constant $\alpha$), to ensure that neither task is completely ignored.
- A supervisor could also use an algorithm that only considers the current allocation. For example, if $H_{exploit} < \lambda * H_{explore}$ for a certain number of time steps, then the supervisor would add an explorer, and visa versa. For the particular random field illustrated in Fig. 7, there was a significant advantage to switching an exploiter to be an explorer sometime between time steps 75 and 100. The relative values of the heuristics for the constant all-exploiter allocation during that simulation are shown in Fig. 8. Notice that if the supervisor had a rule to add an explorer when $H_{exploit} < \lambda * H_{explore}$ for the past $2\tau$ (20) time steps, then the allocation would have been changed to 4 exploiters around time step 97, a near-optimal decision. Many more simulations would have to be run to determine if this algorithm could be effective in general.

## VII. FUTURE EXPERIMENTS

More tests must be conducted to determine if the specific heuristics proposed in this paper are the most informative for a supervisor tasked with making optimal allocation decisions. Although it has been shown that these heuristics do capture many of the intuitive aspects of exploring and exploiting performance, it has not been shown that the recommendations from these heuristics yield optimal or near-optimal dynamic allocation decisions.

A method for determining the scaling parameter $\lambda$ should be investigated further. It is unclear which of the parameters in Table I affect the value of $\lambda$, since some parameters (e.g. $v_{explore}$) are already accounted for in the calculation of $\widetilde{I}(t)$. It also remains to be seen how sensitive a human supervisor is to the scaling factor; if he or she is maximizing $max(H_{exploit}, \lambda * H_{explore})$, for example, then the allocation decisions may not be incredibly sensitive to the choice of scaling.

## REFERENCES

[1] Y. Jin, A. A. Minai, and M. M. Polycarpou, "Cooperative real-time search and task allocation in UAV teams," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 7–12.
[2] G. Ping-an and C. Zi-xing, "Multi-robot task allocation for exploration," *Journal of Central South University of Technology*, vol. 13, pp. 548–551, October 2006.
[3] K. Lerman, C. Jones, A. Galstyan, and M. J. Matarić, "Analysis of dynamic task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 25, pp. 225–241, 2006.
[4] M. J. Matarić, G. S. Sukhatme, and E. H. Østergaard, "Multi-robot task allocation in uncertain environments," *Autonomous Robots*, vol. 14, pp. 255–263, 2003.
[5] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 41, pp. 148–177, 1979.
[6] J. D. Cohen, S. M. McClure, and A. J. Yu, "Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 362, pp. 933–942, May 2007.
[7] D. W. Stephens and J. R. Krebs, *Foraging Theory*. Princeton University Press, 1986.
[8] J. R. Krebs, A. Kacelnik, and P. Taylor, "Test of optimal sampling by foraging great tits," *Nature*, vol. 275, Sept. 1978.
[9] D. Baronov and J. Baillieul, "Search decisions for teams of automata," in *Proceedings of the 47nd IEEE Conference on Decision and Control*, 2008, pp. 1133–1138.
[10] S. J. Payne, G. B. Duggan, and H. Neth, "Discretionary task interleaving: Heuristics for time allocation in cognitive foraging," *Journal of Experimental Psychology: General*, vol. 136, pp. 370–388, 2007.
[11] E. K. P. Chong, C. Kreucher, and A. Hero, "Adaptive sensing via partially observable markov decision process approximations," *Discrete Event Dynamic Systems*, 2009.
[12] N. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. Fratantoni, and R. Davis, "Collective motion, sensor networks, and ocean sampling," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, Jan. 2007.
[13] F. Zhang, D. M. Fratantoni, D. A. Paley, J. M. Lund, and N. E. Leonard, "Control of coordinated patterns for ocean sampling," *International Journal of Control*, vol. 80, pp. 1186–1199, 2007.
[14] W. Burgard, M. Moors, and F. Schneider, *Collaborative Exploration of Unknown Environments with Teams of Mobile Robots*. Springer Berlin / Heidelberg, 2002.