2008 IEEE Swarm Intelligence Symposium
St. Louis MO USA, September 21-23, 2008

Using Scouts to Predict Swarm Success Rate

Antons Rebguns, Richard Anderson-Sprecher, Diana Spears, William Spears and Aleksey Kletsov

***Abstract*—The scenario addressed here is that of a swarm of agents (simulated robots) that needs to travel from an initial location to a goal location, while avoiding obstacles. Before deploying the entire swarm, it would be advantageous to have a certain level of confidence that a desired percentage of the swarm will be likely to succeed in getting to the goal. The approach taken in this paper is to use a small group of expendable robot "scouts" to predict the success probability for the swarm. Two approaches to solving this problem are presented and compared – the standard Bernoulli trials formula, and a new Bayesian formula. Extensive experimental results are summarized that measure and compare the mean-squared error of the predictions with respect to ground truth, under a wide variety of circumstances. Experimental conclusions include the utility of a uniform prior for the Bayesian formula in knowledge-lean situations, and the accuracy and robustness of the Bayesian approach. The paper also reports an intriguing result, namely, that both formulas usually predict better in the presence of inter-agent forces than when their independence assumptions hold.**

***Index Terms*—Scouts, Success rate, Swarm.**

I. INTRODUCTION

SUPPOSE that a swarm of agents (simulated robots) is deployed in an environment with the objective of transporting itself from an initial location to a goal location, while avoiding obstacles. The obstacles and other impediments in the environment could prevent some (or all) of the agents from reaching the goal within a reasonable amount of time, or ever. The environment in which the agents are deployed is assumed to be static, though it may be completely or partially unknown. This environment can be highly unstructured as well, as in Mondada et al. [12]. Furthermore, deployment of the entire swarm is potentially hazardous, e.g., due to the possible loss or corruption of agents – for example, some of the obstacles might contain explosives, robots could fall into inescapable holes as in [3], or there could be crisis environmental dangers as in [18]. In other words, the situation being addressed here requires a preliminary phase of *risk assessment* prior to full swarm deployment. This situation calls for swarm intelligence in the sense of "testing the waters" to quantify the risk before deploying the full swarm.

Like previous related work (see Section II), our approach involves sending out "scout agents" to assess the risk. Only a few scouts are sent from the swarm, which may consist of hundreds of robots. A person/agent ("sender") deploys the scouts, and a person, scout or other agent, or a sensing device ("receiver") counts the fraction of scouts that arrive at the goal successfully. The sender and receiver must be able to communicate with each other – to report the scout success rate, but no other agents require the capability of communicating messages. Based on the fraction of scouts that successfully reach the goal, we apply a formula that predicts the probability that a desired percentage of the entire swarm will get to the goal. Based on this probability, the sender can decide whether or not to deploy the full swarm. Alternatively, based on this probability the sender can decide how many robots to deploy in order to yield a high probability that a desired number of agents will get to the goal. The solution to this swarm risk assessment problem is critical for search-and-rescue and related swarm applications in hazardous environments and/or difficult terrain.

This paper reports two formulas that use robot scouts as "samples" for making predictions regarding the success probability of a swarm. The first approach is the standard Bernoulli trials formula, and the second is a novel Bayesian formula. The paper also reports conclusions regarding the predictive accuracy of these formulas, based on an extensive set of experiments during which parameters were varied methodically. Our measure of predictive accuracy is the mean-squared error of each formula's predictions versus ground truth. Experimental conclusions include the value of a uniform prior for the Bayesian formula in knowledge-lean situations, and the accuracy and robustness of the Bayesian approach. This paper also reports an intriguing result, namely, that both formulas usually predict better in the presence of inter-agent forces than when their independence assumptions hold. Inter-agent forces are useful for initiating and sustaining multi-robot formations while traveling to a target location. We conclude that these formulas, and especially our Bayesian formula, provide extremely practical solutions for solving "the swarm success rate prediction problem" in a wide variety of real-world situations. Furthermore, this paper provides useful advice on selecting the values of controllable parameters in order to help the practitioner apply our approach.

II. RELATED WORK

Greene and Gordon have published a study of red harvest ants that use a group of "patrollers" as scouts before sending out the full swarm of ants [7]. These patrollers establish a path for foraging. Our scouts also act as "patrollers," but they quantify the probability of success for the swarm rather than setting the path. Future work will include merging these two approaches.

Antons Rebguns is with Department of Computer Science, University of Wyoming, Laramie, Wyoming 82071. (e-mail: anton@cs.uwyo.edu).
Richard Anderson-Sprecher is with Department of Statistics, University of Wyoming, Laramie, Wyoming 82071. (e-mail: sprecher@uwyo.edu).
Diana Spears is with Department of Computer Science, University of Wyoming, Laramie, Wyoming 82071. (e-mail: dspears@cs.uwyo.edu).
William Spears is with Department of Computer Science, University of Wyoming, Laramie, Wyoming 82071. (e-mail: wspears@cs.uwyo.edu).
Aleksey Kletsov is with Department of Natural Sciences, Coastal Carolina College, Jacksonville, North Carolina 28546. (e-mail: kletsovA@coastal.cc.nc.us).

Given the detailed quantified predictions of our approach, it can be considered a "probabilistic macroscopic model for swarm robotic systems" in the sense of Lerman, Martinoli and Galstyan [11]. Our approach differs from their Markovian models, on the other hand, because it uses a combination of model and data (the scouts/samples). The combination of model and data also distinguishes our approach from Lyapunov stability methods such as [5]. Stability proofs ensure convergence (success) in the limit as time goes to infinity, but not a *probability* of success.

The task of avoiding obstacles while navigating to a goal is classical in robotics, including in *swarm* robotics, e.g., [4], [6], [8]–[10], [16]. Recently, there has been a growing interest in the use of scout agents for this and related tasks. Scouts are particularly relevant for search-and-rescue applications [14], for remote reconnaissance (www-eng.llnl.gov/tsr/tsr.html), and within the context of swarms they are relevant for distributed sensor networks [19] and swarm "nest assessment" [15]. Like this previous work, our approach also uses scouts for assessing the safety of the environment for the swarm. We have not, however, been able to find any prior research on the topic of sending out a small set of scout agents specifically for the purpose of predicting the *probability* of swarm success. The ability to predict the swarm success rate is fundamental for tasks in which the loss of a substantial fraction of the swarm would be costly or might prevent completion of the task at hand.

III. OUR APPROACH

Our objective is to predict the probability that y or more out of a total of n agents will successfully navigate through an obstacle field and get to a goal location within a time limit. We predict this probability by sending out a sample of k scout agents (which, we assume, are not part of the n swarm agents, although modifying our formula if they are a subset of the swarm is straightforward). For practical reasons, we assume that $k << n$.

A. The Standard Bernoulli Trials Formula

Clearly, the process of predicting the swarm success rate can be considered to be a Bernoulli trials process. The standard formula from [22], which we call P_{Bern}, for predicting the probability of y or more successes in n independent trials is directly applicable:

$$P_{Bern} = P(Y \geq y|\hat{p}) \;=\; \sum_{j=y}^{n} \left(\begin{array}{c} n \\ j \end{array} \right) \hat{p}^j (1-\hat{p})^{n-j}$$

where Y is a random variable. This assumes that k scouts are used to find an estimate $\hat{p} = \frac{Successes}{k}$ of p, where p is the true probability of one agent succeeding in the time limit. We use p as a measure of environmental difficulty.

B. Our Bayesian Formula

A well-known problem of P_{Bern} is that k should be large, perhaps more than 20 scouts, for this to provide a reasonable estimate $\hat{p}$ of p. When k is as small as we assume here (3 to 15 scouts), the variance of P_{Bern} tends to be high, and there is a substantial chance that either all or no scouts will reach the goal, typically leading to poor predictions of swarm behavior. As an example, consider the case of three scouts. There is always a risk that all three will fail, even if the environment is very easy, i.e., tiny samples can be skewed.

To overcome this limitation of P_{Bern}, we provide a Bayesian formula called P_{Bayes}. This formula assumes a prior distribution over the probability p of a single agent getting to the goal. Intuitively, the use of a prior distribution with P_{Bayes} can serve to initialize with prior information (if such information is available), it reduces the variance of the predictions (and this lower variance has been demonstrated experimentally), and it "evens out" the random perturbations of small samples thereby enabling greater accuracy with small k. We use a Beta distribution (pdf) $B(\alpha, \beta)$ for the prior [22]. If the parameters α and β are both 1, then this is the uniform distribution. The distribution is over (0,1).

Recall that our objective is to find the probability of y or more successes out of n agents. We are allowed to observe k scouts, and their performance yields a fraction $\hat{p}$ of successes that is used to predict the true probability p of one agent getting to the goal. Our Bayesian formula is (see the derivation in the Appendix below):

$$P_{Bayes} = P(Y \geq y|\hat{p}) \;=$$

$$\int_0^1 \int_0^p M \cdot z^{y-1}(1-z)^{n-y} p^{r-1}(1-p)^{s-1} dz dp$$

where $M = \{B(r,s) \cdot B(y, n-y+1)\}^{-1}$, $r = k\hat{p} + \alpha$, $s = k(1-\hat{p}) + \beta$, and p takes on all possible values in the outer integral, and z is a variable that ranges from 0 to p, given the particular p chosen by the outer integral. Note that this formula is not the same formula as one would obtain by plugging the posterior mean of p into the formula for P_{Bern}. It is the posterior mean of the probability that y or more scouts will reach the goal, assuming a Beta prior for p with parameters α and β. The inner integral gives the probability of y or more successes out of n trials for a given p, weighted by the prior for p. The form of the integrand may not be intuitive because it replaces the usual sum of Binomial probabilities with an equivalent incomplete Beta form (see [1] for details). The outer integral averages over possible values p of environmental difficulty, which ranges from 0 to 1.

P_{Bayes} is implemented in our simulation using numerical methods, i.e., Simpson's rule for double integrals and Gaussian quadrature [2], [13], [22]. The implementation of this formula was rigorously tested against hand-calculated values, and found to be accurate and correct within expected bounds of precision.

IV. OUR SWARM ROBOTICS SIMULATOR

To visualize and evaluate our solution, we have designed and implemented a simulation (see Fig. 1) consisting of a distributed, decentralized swarm of agents (the dots), a goal location (square), and a set of obstacles (circles). The locations

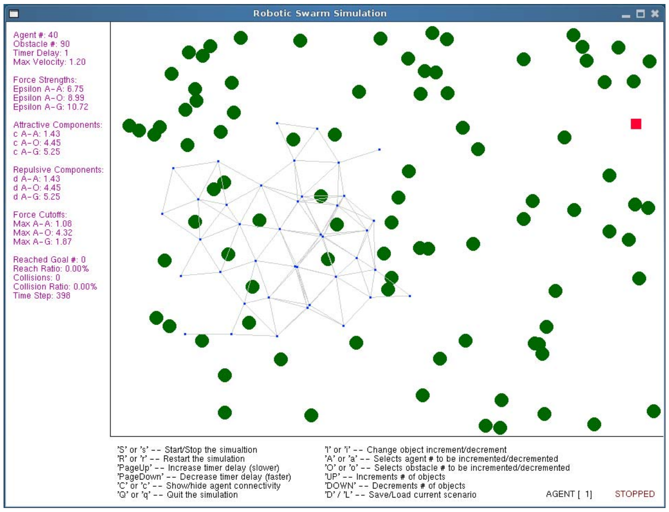

Fig. 1. Swarm simulation screenshot

of all of these entities can be varied. Agents can sense the goal at any distance, and obstacles within range $1.5R$, where R is 50 pixels. Their control algorithm is based on *physicomimetics* [17], although our approach is not dependent on this particular control algorithm and potentially can be used with any other swarm control algorithm. In physicomimetics, agents experience virtual (artificial) attractive forces from a goal and virtual repulsive forces from obstacles. Our variant of the Lennard-Jones (LJ) force between i and j is:

$$F_{i,j} = 24\epsilon \left[\frac{2dR^{12}}{r^{13}} - \frac{cR^6}{r^7} \right]$$

where the parameter ϵ affects the strength of the force between the objects, r is the actual distance between i and j, R is the desired distance between i and j, and c and d control the relative balance between attractive and repulsive forces, respectively. The LJ force law was selected because it is extremely effective for robots navigating around obstacles to get to a goal. We use optimal parameter settings for this force law that were evolved for the task by Hettiarachchi [8]. To use the LJ force law, agents sense the range and bearing to nearby obstacles as well as to the goal location.

Initially, all agents begin at uniformly randomly chosen locations within a square, approximately 1,000 pixels from the goal. The simulation ends after t = 1,500 time steps. We can run this simulation in two modes: one where the agents are completely independent of each other, and another where there are virtual LJ inter-agent forces. Fig. 1 shows the simulation with inter-agent forces. In this case, the agents stay in a lattice formation while navigating to the goal location.

V. EXPERIMENTAL SETUP

A. The Performance Metrics

We use the mean squared error (MSE) for our performance metric here because it includes both the error and the variance of the error, and is therefore highly informative. The "error" is the difference between the estimate and the truth. If we assume that X is the true value and $\hat{X}$ is an estimate of X, then the MSE is defined as:

$$MSE(\hat{X}, X) = E[(\hat{X} - X)^2]$$

where E denotes the expected value.

To measure the MSE, we need a form of "ground truth." Our ground truth, which we call P_{Truth}, is the fraction out of 1,000 runs of the simulation in which y or more of the n agents get to the goal. To compare with ground truth, our performance metrics are $MSE(P_{Bern}, P_{Truth})$ and $MSE(P_{Bayes}, P_{Truth})$.

B. Parametric Experimental Design

We wish to compare the performance metrics $MSE(P_{Bern}, P_{Truth})$ and $MSE(P_{Bayes}, P_{Truth})$ under a wide variety of parametric variations. Our comparisons are fair, i.e., when comparing P_{Bern} or P_{Bayes} with P_{Truth}, all three always use the same values of all parameters.

Our experimental objective is to vary the experimental parameters and measure how MSE increases or decreases as a function of the variations. A factorial experimental design has been performed. As in a typical factorial experiment design, we vary one parameter at a time and consider *all* combinations of *all* parameter values. Factorial experiments are notoriously computationally lengthy. To make our experiments tractable, we select only a few representative values for each parameter to vary. The following list enumerates the parameters (the **independent variables**) that are varied in the experiment:

- **Total number of agents in the swarm.** This is n and its values are 100, 300, and 500.
- **Number of scouts.** This is k, the number of scouts employed. The values of k are 3, 5, 10, and 15.
- **Desired number of successes.** The variable y is the desired number out of the n swarm agents that are expected to reach the goal. It varies exhaustively from 1 to n.
- **Environmental difficulty.** We also vary p, the "true" probability of one agent succeeding, which is found using 1,000 simulation runs of n agents. Five environments were hand-crafted (by choosing the goal location, and the number, sizes, and locations of obstacles, and the location of the starting square for the agents) in order to get approximately the following values of p: 0.1, 0.3, 0.5, 0.7, and 0.9.
- **Strength of the prior distribution for P_{Bayes}.** Recall that P_{Bayes} uses a Beta prior probability distribution. The "strength" of this prior is quantified as $\frac{1}{\sigma^2}$, where σ^2 denotes the variance. In other words, a lower variance yields a "stronger" prior.
- **Correctness of the prior distribution for P_{Bayes}.** "Correctness" is the degree to which the prior is well-matched to the truth (p). Here, correctness is measured as the absolute value of the difference between the mean of the prior distribution and the true value of p.[1]

[1]Correctness can be measured in our experimental evaluations, but it is not known by a practitioner running the robots in the real world.

- **Independence of the agents.** This parameter is Boolean-valued. In the first case, the agents are assumed to be completely independent of each other. Note that this independence assumption is inherent in both formulas P_{Bern} and P_{Bayes}. To model many real-world situations, we also explore a second case in which we use the LJ inter-agent forces that were optimized via evolutionary algorithms by Hettiarachchi [8]. This second case violates the independence assumptions of both formulas.

The methodology that was used to vary the strength and correctness (by varying α and β) of the priors is described in [1]. For our experiments, we have selected seven priors that vary along the strength and correctness dimensions: (1) the uniform (**weakest**) prior, (2) a **weak, almost correct prior for** p **= 0.1**, (3) a **strong, almost correct prior for** p **= 0.1**, (4) a **weak, almost correct prior for** p **= 0.5**, (5) a **strong, almost correct prior for** p **= 0.5**, (6) a **weak, almost correct prior for** p **= 0.9**, and (7) a **strong, almost correct prior for** p **= 0.9**. Note that a prior that is correct for a particular value of p will be incorrect for any *other* value of p. For example, a prior that is correct for p = 0.1 will be incorrect if the true p is 0.3, and will be even more incorrect if the true p is 0.5 or especially 0.9. This incorrectness is exacerbated if the prior is strong. Also, the reason for "almost" correct is that it is usually too hard to design environments that have precisely a given p level of difficulty, so we approximate as closely as possible.

C. The Experimental Algorithms

Next, we describe our algorithms used in the experiments. Each experiment measures $MSE(P_{Bern}, P_{Truth})$ and $MSE(P_{Bayes}, P_{Truth})$ with a single choice of values for all parameters, for multiple runs.

The algorithm for P_{Truth} was described above. The following is the algorithm for P_{Bayes} and P_{Bern}:

1) Using the k scout agents obtain $\hat{p}$, which is an estimate of p.
2) Apply the implemented formula for $P(Y \geq y|\hat{p})$, using the $\hat{p}$ obtained from Step 1.

The expectation E is calculated as an average over 1,000 runs. For each run, we randomly vary the agent locations within the starting square. The **dependent variable** MSE is calculated as follows. Let P_B be P_{Bern} or P_{Bayes}. Then

$$Bias = E(P_B) - P_{Truth}$$
$$Variance = E[(P_B - E(P_B))^2]$$
$$MSE = E[(P_B - P_{Truth})^2] = Bias^2 + Variance$$

VI. EXPERIMENTAL RESULTS AND CONCLUSIONS

Because we have performed a factorial experiment design, the results are voluminous (i.e., there are hundreds of graphs). There is only room here to show a few of the most interesting and representative results. In the graphs, the horizontal axis is one of the independent variables and, unless otherwise stated, the vertical axis is the MSE. Each curve shows the MSE estimated by averaging over 1,000 runs.

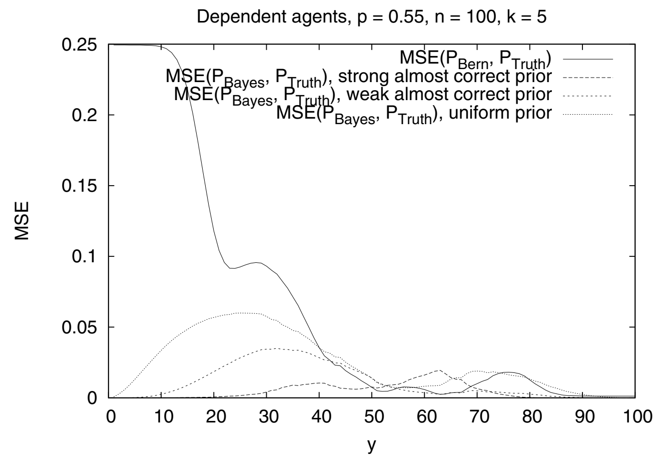

Fig. 2. Graph of MSEs with varying priors for P_{Bayes}, with p = 0.55, n = 100, k = 5, and inter-agent forces

A. Results and Conclusions

Here, we present results with both independent and dependent agents. The case of independent agents is practical for some search and rescue problems, as well as chemical plume tracing applications where the robots use *anemotaxis* (e.g., [20], [21]). The more practical case, also included here, is that of dependent swarm agents. In both cases, scout behavior is representative of swarm behavior. I.e., when the swarm consists of independent agents, the scouts are also independent; when the swarm consists of dependent agents, the scouts are also dependent.

Our experimental results are presented next. For all the graphs, we vary the parameter of interest and hold the others (except y) constant. When y is not on the horizontal axis, each curve is an average over all values of y. The following results/conclusions hold for both the independent and dependent agent cases:

1) Both formulas are usually good predictors. Their accuracy can be seen in all of our graphs.
2) Observations about prior distribution selection for our Bayes formula are:
   - It is better to have a correct rather than an incorrect prior.
   - If the prior is incorrect it is better to be weak rather than strong. Uniform (the weakest) is the best incorrect prior.
   - If the prior is correct it is better to be strong than weak.

   For example, Fig. 2 confirms the first and third observations over most values of y, where y is varied along the horizontal axis. Overall, we have found that a strong, correct prior is optimal, and a uniform prior is the best choice when prior knowledge is lacking.
3) P_{Bayes} with a uniform prior is usually better (i.e., has a lower MSE) than P_{Bern}, and is often much better. This conclusion has significant practical implications. It means that even in the case when prior knowledge is absent, P_{Bayes} typically outperforms P_{Bern}. All graphs

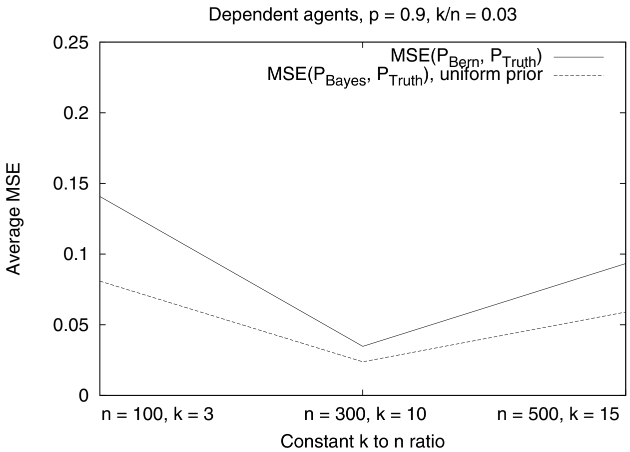

Fig. 3. Graph of MSE with varying n and k, $\frac{k}{n}$ = 0.03, p = 0.9, uniform prior for P_{Bayes}, and inter-agent forces

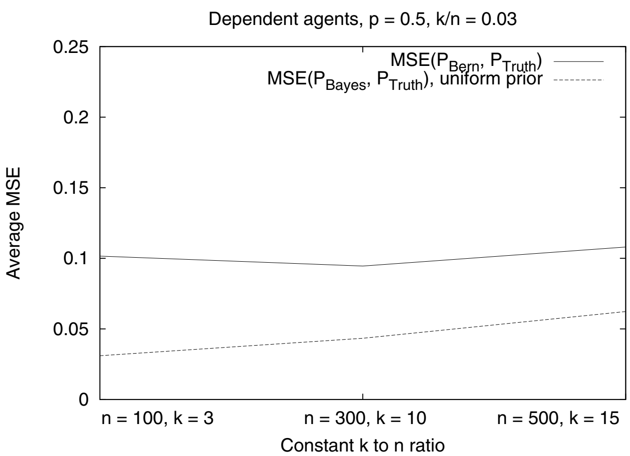

Fig. 4. Graph of MSE with varying n and k, $\frac{k}{n}$ = 0.03, p = 0.5, uniform prior for P_{Bayes}, and inter-agent forces

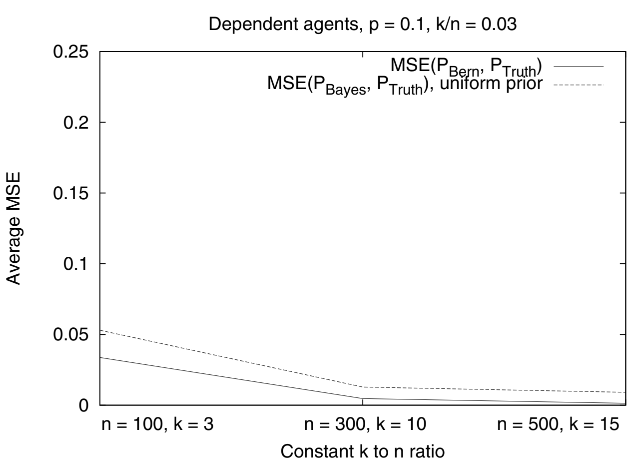

Fig. 5. Graph of MSE with varying n and k, $\frac{k}{n}$ = 0.03, p = 0.1, uniform prior for P_{Bayes}, and inter-agent forces

in this paper that compare P_{Bayes} with P_{Bern} provide evidence to support this conclusion.

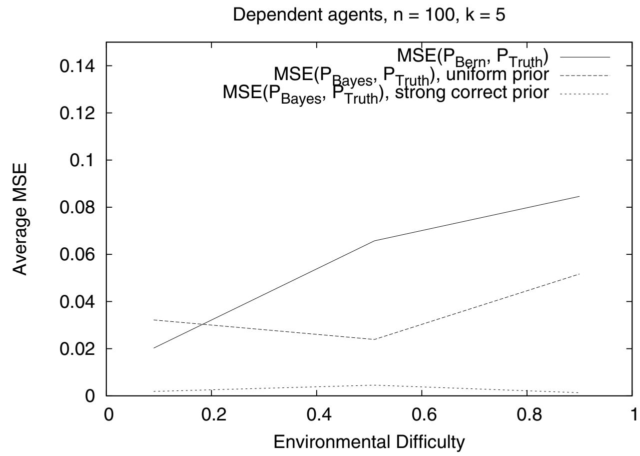

Fig. 6. Graph of $MSE(P_{Bern}, P_{Truth})$ and $MSE(P_{Bayes}, P_{Truth})$, with inter-agent forces, as environmental difficulty varies

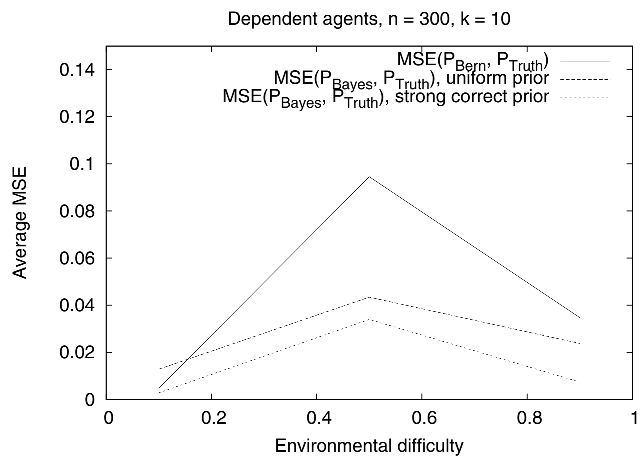

Fig. 7. Graph of $MSE(P_{Bern}, P_{Truth})$ and $MSE(P_{Bayes}, P_{Truth})$, with inter-agent forces, as environmental difficulty varies

4) Holding the ratio $\frac{k}{n}$ constant and increasing the swarm size n has negligible effect on predictability. Figs. 3, 4, 5 show what the results usually look like for a dependent swarm and easy ($p = 0.9$), moderate ($p = 0.5$), and hard ($p = 0.1$) environments, respectively.
5) P_{Bayes} is "robust" across wide variations in environmental difficulty. By "robust" we mean that the MSE does not substantially increase as the environmental difficulty is varied. P_{Bern} is less robust than P_{Bayes}. In particular, when averaged over all environments (which is an uncontrollable parameter), P_{Bayes} appears to be preferable. P_{Bern} shows a lack of robustness that can make it barely acceptable in moderate to easy environments. Figs. 6 and 7 show two representative examples for the dependent agents case.

Finally, we wish to point out a prevalent trend related to variations in the value of y. Note the bimodal curves in Fig. 8, which tend to be a common characteristic of our error curves when y is varied along the horizontal axis. To understand this bimodal pattern, it is useful to look at the graph of P_{Bayes} versus P_{Truth} that was used to create one of the curves

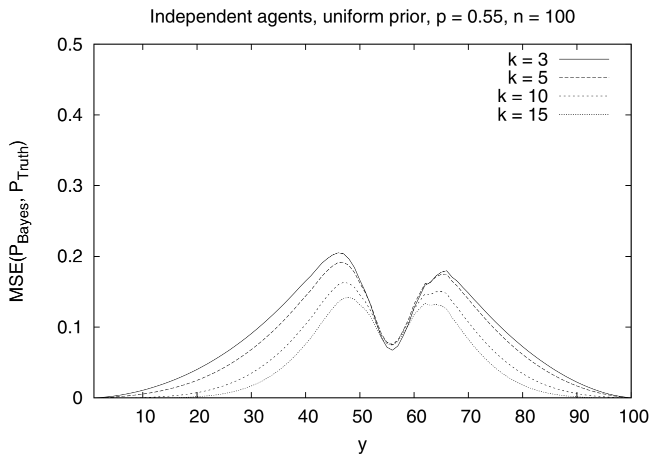

Fig. 8. Graph of $MSE(P_{Bayes}, P_{Truth})$ with p = 0.55 and n = 100

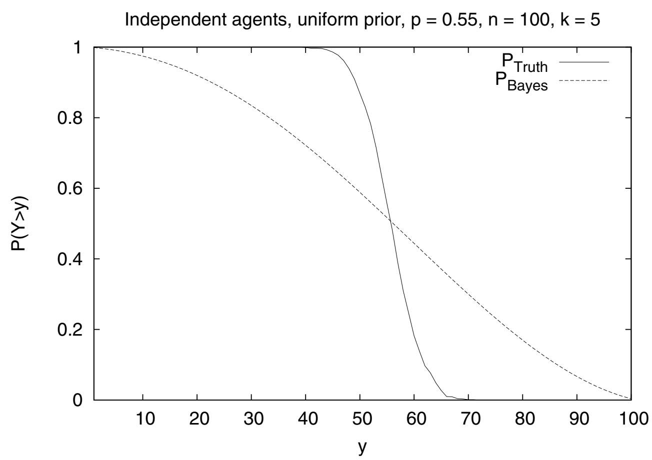

Fig. 9. Graph of P_{Bayes} and P_{Truth} with p = 0.55, n = 100, and k = 5

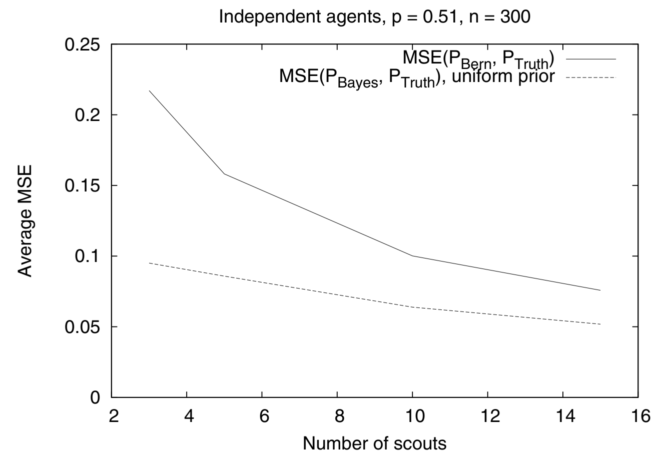

Fig. 10. Graph of MSEs with p = 0.51, n = 300, k ranging from 3 to 15, with *no* inter-agent forces

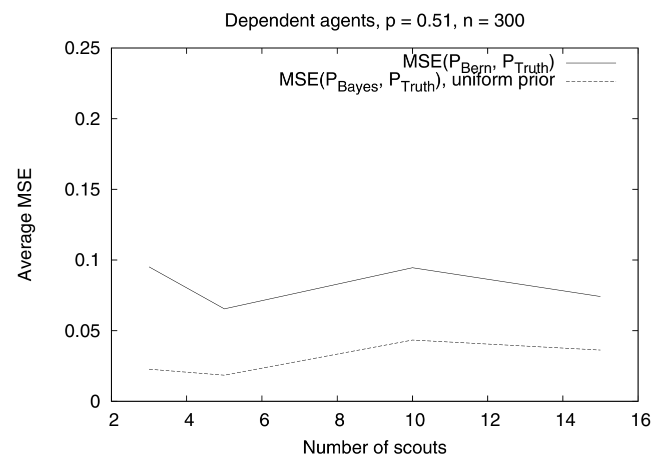

Fig. 11. Graph of MSEs with p = 0.51, n = 300, k ranging from 3 to 15, with inter-agent forces

in Fig. 8. Fig. 9 provides the intuition for the bimodal pattern. Compare the curve for P_{Bayes} with the curve for P_{Truth} in Fig. 9. What we see is that there is an abrupt shift in ground truth (P_{Truth}) probability from 1 to 0, analogous to a sigmoid function (but reversed), around y = 55. We call this the "phase transition." In particular, if $y < np$, then y or more agents are guaranteed of getting to the goal – because the environment (captured by p) supports such success. However if $y > np$, then it is nearly impossible (probability close to 0) for y or more agents to get to the goal. P_{Truth} accurately reflects this phase transition. However the curve for P_{Bayes} very roughly approximates the transition, i.e., rather than an abrupt drop in the curve it shows a smaller slope in going from a 1 to 0 probability of success. This explains the curves for MSE. Near the transition point where $y = np$, the curves for P_{Truth} have an abrupt phase shift but the curves for P_{Bayes} have a more gradual shift, which causes the bimodal errors around the point $y = np$. The same phenomenon occurs in all the graphs comparing P_{Bern} with P_{Truth} as well. Although our graphs show results for only one particular set of parameters, our factorial experiments have confirmed that this characteristic pattern holds regardless of *all* the other parameter settings, when y is the independent variable.

B. Independent Versus Dependent Agents

Perhaps the most interesting result we have found is that although our two formulas make independence assumptions, when these assumptions are violated the predictions of both formulas usually improve! Figs. 10 and 11 illustrate this phenomenon. All parameter values are identical in these two figures, other than inter-agent (in)dependence (for both the scouts and the swarm). This exciting result about the behavior with dependent agents implies that the P_{Bern} and P_{Bayes} formulas are especially applicable in real-world swarm situations requiring inter-agent coordination.

In these figures, one can also see the usual result that P_{Bayes} is a better predictor than P_{Bern}, despite P_{Bayes}'s use of a uniform prior.

VII. Advice to the Swarm Practitioner

This section focuses on advice to the swarm practitioner in the case of inter-agent forces. The maximum value seen in our graphs (averaged over y) of $MSE(P_{Bern}, P_{Truth})$ is 0.14, whereas the maximum value seen in our graphs (again, averaged over y) of $MSE(P_{Bayes}, P_{Truth})$ is 0.08. This implies that P_{Bern} gives nearly twice the average mean squared error in the worst case. Detailed experiments show that it is often the error that is to blame, and not just a high variance, for causing P_{Bern} to be less robust. Therefore, if only a few expendable scouts (≤ 15) are available, and the practitioner has a choice of formulas, our experimental results indicate that P_{Bayes} is preferable both in terms of robustness (with a uniform prior) and in terms of optimality (if a strong close-to-correct prior can be chosen).

Consider other practical, real-world implications of our experimental conclusions. First, in knowledge-lean situations P_{Bayes} with the uniform prior is an excellent choice for robustness. Second, in situations where the prior can be reasonably estimated, P_{Bayes} with a strong (almost) correct prior provides the best performance. Third, both formulas are scalable with respect to increasing swarm size (in terms of MSE), if the scouts-to-swarm ratio is held constant at 0.03, which seems to be a reasonable ratio. Finally, both formulas are reasonable for real-world swarm success probability predictions, though P_{Bayes} is overall better.

VIII. SUMMARY AND FUTURE WORK

We have presented a very useful approach for employing scouts to predict the success rate of a swarm in avoiding obstacles and getting to a goal location. Such an approach is especially pertinent in difficult or dangerous situations, when sending the entire swarm at once is inadvisable. Our approach is practical and effective under a wide range of conditions, and is especially useful for swarms of interacting agents.

Future work will include transitioning our approach to a swarm of real robots, and improving the theory. We also plan to record successful scout paths, and to design strategies for increasing the success rate when a low rate is predicted.

REFERENCES

[1] Anderson-Sprecher, R., Rebguns, A., Spears, D., and Kletsov, A. 2007. Predicting the success rate of a swarm. University of Wyoming Technical Report.

[2] Cheney, W. and Kincaid, D. 2007. *Numerical Mathematics and Computing, 5th Edition*. Brooks Cole Publishers.

[3] Dorigo M., Tuci E., Trianni V., Gross R., Nouyan S., Ampatzis C., Labella H.T., O'Grady R., Bonani M., and Mondada F. 2006. SWARM-BOT: Design and implementation of colonies of self-assembling robots. In *Yen G.Y., Fogel D.B., editors, Computational Intelligence: Principles and Practice, IEEE Computational Intelligence Society, NY*; 103–135.

[4] Fasola, J., Rybski, P., and Veloso, M. 2005. Fast goal navigation and obstacle avoidance using a dynamic local visual model. In *Brazilian Symp. of Art. Int.*

[5] Gazi, V. and Passino, K. 2002. Stability analysis of swarms. In *Proc. of ACC'02*; 539–557.

[6] Gilboa, A., Meisels, A., and Felner, A. 2006. Distributed navigation in an unknown physical environment. In *Proc. of AAMAS'06*; 553–560.

[7] Greene, M. J., and D. M. Gordon. 2007. How patrollers set foraging direction in harvester ants. American Naturalist 170: 943–948.

[8] Hettiarachchi, S. 2007. *Distributed evolution for swarm robotics.* Ph.D. dissertation. University of Wyoming.

[9] Hettiarachchi, S., and Spears, W. 2005. Moving swarm formations through obstacle fields. In *International Conference on Artificial Intelligence, Vol. 1*; 97–103.

[10] Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research 5(1): 90–98.*

[11] Lerman, K., Martinoli, A., and Galstyan, A. 2004. A review of probabilistic macroscopic models for swarm robotic systems. In *Proc. of SAB'04 Swarm Robotics Workshop*.

[12] Mondada, F., Gambardella, L.M., Floreano, D., Nolfi, S., Deneubourg, J-L., and Dorigo, M. 2005. The cooperation of swarm-bots: Physical interactions in collective robotics. IEEE Robotics & Automation Magazine, Vol. 12, No 2: 21–28.

[13] Press, W., Teukolsky, S., Vetterling, W., Flannery, B. 1992. *Numerical Recipes in C, 2nd Edition.* Cambridge University Press.

[14] Rybski, P., Burt, I., Drenner, A., Kratochvil, B., McMillen, C., Stoeter, S., Stubbs, K., Gini, M., and Papanikolopoulos, N. 2001. Evaluation of the scout robot for urban search and rescue. *AAAI Workshop.*

[15] Sahin, E. and Franks, N. 2002. Simulation of nest assessment behavior by ant scouts. In *Proc. of ANTS Workshop*; 274–282.

[16] Spears, D., Kerr, W., and Spears, W. 2006. Physics-based robot swarms for coverage problems. *Int'l Journal on Intelligent Control and Systems 11(3): 124–140.*

[17] Spears, W., Spears, D., Hamann, J., and Heil, R. 2004. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots 17(2-3): 137–162.*

[18] Tatomir, B. and Rothkrantz, L. 2006. Ant based mechanism for crisis response coordination. In *Proc. of ANTS'06 Workshop*; 380–387.

[19] Tikanmaki, A., Haverinen, J., Kemppainen, A., and Roning, J. 2006. Remote-operated robot swarm for measuring an environment. *International Conference on Machine Automation.*

[20] Zarzhitsky, D., Spears, D., Thayer, D., and Spears, W. 2004. A fluid dynamics approach to multi-robot chemical plume tracing. In *3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*; 1476–1477.

[21] Zarzhitsky, D., Spears, D., and Spears, W. 2005. Distributed robotics approach to chemical plume tracing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*; 4034–4039.

[22] Zwillinger, D. 2003. *CRC Standard Math. Tables and Formulae, 31st Edition.* CRC Press.

APPENDIX

This appendix provides our derivation of the P_{Bayes} formula. Here, we employ the common statistics convention of letting the argument determine which probability function is meant by f, i.e. f varies depending on the context. Also, we use π rather than f for the *prior* distribution, as is traditional in the statistics literature.

Let X be a binomial random variable, where $X|p \sim B(n,p)$ and p is the probability of one robot getting to the goal and n is the total number of robots. Let $\hat{p}$ be an estimate of p based on k trials. Then $k\hat{p}|p \sim B(k,p)$, and these variables are independent. From the Bernoulli trials formula, we have:

$$f(x|p) = \begin{pmatrix} n \\ x \end{pmatrix} p^x (1-p)^{n-x}$$

$$f(\hat{p}|p) = \begin{pmatrix} k \\ k\hat{p} \end{pmatrix} p^{k\hat{p}} (1-p)^{k(1-\hat{p})}$$

Assume that $p \sim Beta(\alpha,\beta) = B(\alpha,\beta)$, so that the prior distribution $\pi(p) = \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}$. We can now calculate $P(X \geq y|\hat{p})$, abbreviated $P(y)$. Let us begin with the probability of *exactly* x successes, given $\hat{p}$:

$$f(x|\hat{p}) = \frac{f(x,\hat{p})}{f(\hat{p})} = \frac{\int_0^1 f(x,\hat{p},p)dp}{\int_0^1 f(\hat{p},p)dp} = \frac{\int_0^1 f(x,\hat{p}|p)\pi(p)dp}{\int_0^1 f(\hat{p}|p)\pi(p)dp}$$

by the definition of conditional probability and the product rule. Now, expand the numerator:

$$\int_0^1 f(x,\hat{p}|p)\pi(p)dp = \int_0^1 f(x|p)f(\hat{p}|p)\pi(p)dp$$

by the conditionalized product rule. Substitution yields:

$$= \int_0^1 \binom{n}{x} p^x(1-p)^{n-x} \binom{k}{k\hat{p}} p^{k\hat{p}}(1-p)^{k(1-\hat{p})}\times$$

$$\frac{1}{B(\alpha,\beta)}p^{\alpha-1}(1-p)^{\beta-1}dp$$

Then, expand the denominator:

$$\int_0^1 f(\hat{p}|p)\pi(p)dp =$$

$$\int_0^1 \binom{k}{k\hat{p}} p^{k\hat{p}}(1-p)^{k(1-\hat{p})}\frac{1}{B(\alpha,\beta)}p^{\alpha-1}(1-p)^{\beta-1}dp =$$

$$\binom{k}{k\hat{p}}\frac{1}{B(\alpha,\beta)}\int_0^1 p^{k\hat{p}+\alpha-1}(1-p)^{k(1-\hat{p})+\beta-1}dp =$$

$$\binom{k}{k\hat{p}}\frac{B(k\hat{p}+\alpha,k(1-\hat{p})+\beta)}{B(\alpha,\beta)}$$

Because the denominator is free of x, we will temporarily refer to it as D. Now, for exactly x success we have:

$$f(x|\hat{p}) = \frac{\int_0^1 f(x,\hat{p}|p)\pi(p)dp}{D}$$

$$Thus, \quad P(y) = \frac{1}{D}\sum_{x=y}^{n}\int_0^1 f(x,\hat{p}|p)\pi(p)dp =$$

$$\frac{1}{D}\int_0^1 \sum_{x=y}^{n}\binom{n}{x} p^x(1-p)^{n-x}\binom{k}{k\hat{p}} p^{k\hat{p}}(1-p)^{k(1-\hat{p})}\times$$

$$\frac{1}{B(\alpha,\beta)}p^{\alpha-1}(1-p)^{\beta-1}dp =$$

$$\frac{1}{D}\binom{k}{k\hat{p}}\frac{1}{B(\alpha,\beta)}\int_0^1 \sum_{x=y}^{n}\binom{n}{x} p^x(1-p)^{n-x}\times$$

$$p^{k\hat{p}}(1-p)^{k(1-\hat{p})}p^{\alpha-1}(1-p)^{\beta-1}dp$$

Now apply the relation between the cumulative Binomial probability and the incomplete Beta function:

$$\sum_{x=y}^{n}\binom{n}{x} p^x(1-p)^{n-x} = I_p(y,n-y+1) =$$

$$\frac{1}{B(y,n-y+1)}\int_0^p z^{y-1}(1-z)^{n-y}dz$$

and therefore

$$P(y) = \frac{1}{D}\binom{k}{k\hat{p}}\frac{1}{B(\alpha,\beta)B(y,n-y+1)}\times$$

$$\int_0^1\int_0^p z^{y-1}(1-z)^{n-y}p^{k\hat{p}+\alpha-1}(1-p)^{k(1-\hat{p})+\beta-1}dzdp$$

Returning to D,

$$\frac{1}{D}\binom{k}{k\hat{p}}\frac{1}{B(\alpha,\beta)B(y,n-y+1)} =$$

$$\frac{\binom{k}{k\hat{p}}\frac{1}{B(\alpha,\beta)B(y,n-y+1)}}{\binom{k}{k\hat{p}}\frac{B(k\hat{p}+\alpha,k(1-\hat{p})+\beta)}{B(\alpha,\beta)}} =$$

$$\frac{\frac{1}{B(y,n-y+1)}}{B(k\hat{p}+\alpha,k(1-\hat{p})+\beta)} =$$

$$\frac{1}{B(y,n-y+1)B(k\hat{p}+\alpha,k(1-\hat{p})+\beta)}$$

Therefore,

$$P(y) = \frac{\int_0^1\int_0^p z^{y-1}(1-z)^{n-y}p^{k\hat{p}+\alpha-1}(1-p)^{k(1-\hat{p})+\beta-1}dzdp}{B(y,n-y+1)B(k\hat{p}+\alpha,k(1-\hat{p})+\beta)}$$

When simplified, this becomes our formula in the main body of the paper:

$$P(y) = \int_0^1\int_0^p M\cdot z^{y-1}(1-z)^{n-y}p^{r-1}(1-p)^{s-1}dzdp$$

where

$$M = \{B(r,s)\cdot B(y,n-y+1)\}^{-1}$$

$$r = k\hat{p}+\alpha \quad and \quad s = k(1-\hat{p})+\beta$$