Options in Swarm Robotics.

Raihan Seraj
McGill University, Montreal

***Abstract*—We study the problem of Reinforcement Learning in Robot Swarms. Robotic swarms are useful in several activities such as firefighting, construction, surveillance, sports coverage etc. In all these activities, swarms add robustness, adaptivity and flexibility (reconfigurability) when compared with single robots. Most of these activities require a coordinated behavior by all robots in the swarm. This coordinated behaviour can in several cases be represented by the trajectories of physical locations (configuration trajectories) of the robots in the swarm. In this paper, we only consider this case. Planning optimal configuration trajectories for swarms for various tasks is a challenging problem. One way of overcoming this is to explore reinforcement learning based algorithms. In this work, we study adapted versions of conventional single agent RL algorithms for swarm robotics. We conduct numerical studies in simulated environments and provide preliminary suggestions based on these results.**

***Index Terms*—Reinforcement Learning, Swarms, Multi agent Systems.**

I. Introduction

THE growing importance of swarms in robotics has been inspired by the existence of swarms in nature specially from the interaction of social insects, mammals. The collective behavior of swarms presents greater flexibility ,robustness in terms of certain tasks that require complex behavior without the need for any external coordination . In robotics,swarm intelligence has been applied for tasks that have to be accomplished in large or unstructured environments where no proper infrastructure that can be used to control the robot exists. Such tasks include underwater or extraterrestrial planetary exploration, surveillance, demining, search and rescue,fire fighting etc. Distributed control algorithms for swarms can be classified into two categories which is analogous to the motion in fluids described using either the Lagrangian framework or the Eulerian framework. The Lagrangian framework for swarms considers each of the agents individually where each of the agent's trajectory is generated separately. In contrast the Eulerian framework deals with the collective properties of the swarms where its density distribution is controlled. In this paper we adopt the Eulerian framework where we study the application of Reinforcement Learning for collective learning of a target distribution.

A. Motivation

Reinforcement learning has been widely used for behavior generation in robotics and provides appealing approach for robots to learn new tasks. Both model based and model free approaches had been adapted for solving a wide range of complex tasks in robotics. Model based reinforcement learning allows artificial agents to build an internal model of the environment and plan within it in response to detected environmental changes. On the contrary Model-Free reinforcement learning allows the agents to learn an optimal policy where no proper model of the environment exists.The agent progressively acquires cached estimates of the long-run values of circumstances and actions from retrospective experience. This paper formulates a multi-agent reinforcement learning approach for robotic swarms where the agents discover solutions of their own using learning. Our work has been motivated from the probabilistic and distributed control of large scale swarms where optimal solution for acquiring target distribution shapes exists using planning approaches. We try to extend this idea for the task of acquiring a target distribution which can change over time. We start of reproducing the solution methods as mentioned in the paper and study the case where learning framework can be used for the agents to mimic a static target shape and proceed on proposing an extended framework where similar notion can be used for mimicking the shapes that can change with time.

II. Base Model

A. Notation

Our model is an adaptation of the model given in [1] Let $\mathfrak{S}$ be a swarm of robots with $|\mathfrak{S}| = \mathsf{N} \sim 10^3 - 10^6$. Let $p^{(n)} \in \mathbb{D} \subset \mathbb{R}^d$ denote the position of the n^{th} robot/agent, where $\mathbb{D}$ is the domain in which $\mathfrak{S}$ operate. Without loss of generality, we restrict our attention to $d = 2$, i.e., two dimensional Euclidean spaces. Let ξ_t^{N} denote the empirical distribution of the swarm at time t. This distribution is defined in terms of a discretization, where the domain, $\mathbb{D}$, is divided into $\mathcal{B}$ bins. Similar to the theory of particle filters, let $\hat{\xi}_t$ denote the limiting ($\mathsf{N} \to \infty$) density associated with this swarm.

B. Problem Formulation

The problem of probabilistic swarm guidance (PSG) has been addressed in [2]. We select the approach by [1], wherein the swarm is controlled using a non-stationary policy. The transition matrix associated with this policy is generated by solving a linear program. The inputs required for this linear program are the current and target distributions, the cost matrix for transitions and an optional constraint matrix to specify regions (bins) that are not to be visited. A closed form expression for this matrix has been provided in [1]. Once the transition matrices are determined, the actual low-level controls necessary to achieve this transition are computed using a separate collision-free trajectory computation algorithm.

III. Extension I: Option based Controls

A. Problem Formulation

This problem is almost identical to that described in section II-B. However, in this model, we restrict the action space for each agent in the swarm to movement to a local neighborhood. Each agent can only move n_s steps in a Manhattan-distance like manner at each time step. As described in section [1].They have considered the swarm guidance problem as a hierarchical one, where the lower level actions are computed by a separate collision-free motion planner.

B. Solution Approach

Let π_Ω be a policy defined over options, $\omega \in \Omega$, where Ω is the set of all available options. We use the Option terminology as defined in [3], where they use parametrized intra-option policies. We, instead, use local shortest-path algorithms as our intra-option policies. Thus, in our case, the number of options, i.e., $|\Omega| = \mathcal{B}$. Further, $\pi_\Omega = M_s$, i.e., given the current bin, b, of an agent, $a \in \{0, dots, \mathsf{N}\}$, $\pi^a_\Omega(b) \sim M_s[b, :]$. Let $\omega \in \{0, \ldots, \Omega\}$ be the selected option for agent a. Let b_ω be the corresponding bin to ω. Then the intra-option policy for agent a, at any state, $s \in \{0, \ldots, \mathcal{B}\}$ is $\pi^a_\omega(s) = \min(n_s, D_M(s, \omega))$. Agent a's next bin position is then computed using a translation by $\pi^a_\omega(s)$ steps. This option, ω, is terminated, when agent a's current bin equals the bin indicated by b_ω. We call SM the matrix that can be obtained based on initial configuration for the swarms using the solution described in [1].

C. Algorithm 1

1) Find SM based on initial configuration. Mention that we are doing HMC. This is not optimal in terms of cost and Communicate SM as the option policy to all agents
2) For Loop over time steps:
   a) At each time step, each agent currently not executing an option, selects an option using the option policy. Other agents in the midst of executing their options, continue to do so. (We follow the call and return model).
   b) Each agent executes one step of its intra-option policy.
   c) Each agent checks for the termination condition of its current option.

IV. Experimental Results

In order to find whether the solution proposed in the paper is reproducible, we start with the experiments that were mentioned in the paper. By taking an image of a simple triangle convert it to a binary image and get the RGB values of the pixels. We then convert the RGB values into a vector and normalize them in order to get the target distribution. We set the number of particles to 5000 and their initial configuration is set at random, matrix is calculated initially using the planning solution mentioned in [1]. For this case hellinger distance is used in order to compute the cost matrix. We then used a more complicated shape, such as a picture of the Eiffel Tower and check whether the swarms can form the required shapes.

Fig. 1. Ground Truth Image of the Eiffel Tower

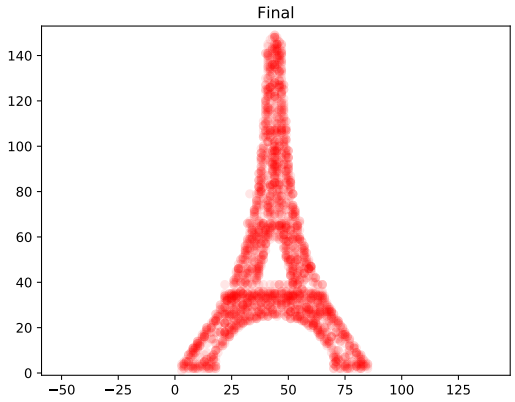

Fig. 2. Swarm Distribution Obtained with the planning solution

We next focus on extend this idea towards the options framework.The solution proposed in the paper allows the swarms to move from one place to another almost instantaneously, with no local restriction imposed on it. However in our options framework we restrict the movement of the agent and test it by restricting movements to 2,5 and 6 steps.We tested different points where the options are terminated. The resulting shape formation is more realistic in the sense that it creates a 'wobble' effect which is practical in swarms as they will tend to keep moving slightly between themselves thereby maintaining the overall distribution. We again tried with multiple shapes with the options framework, and the swarms could effectively replicate the shapes. The following figure shows the results for the formation of a triangle with options framework where we restrict the movement of the agents to 2 steps.

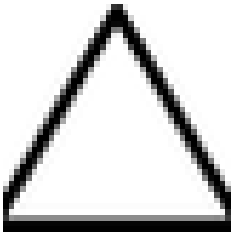

Fig. 3. Ground Truth Image of triangle

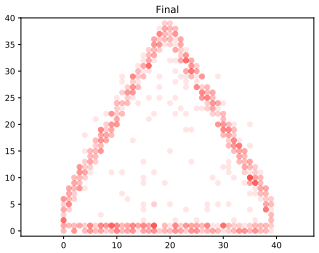

Fig. 4. Swarm Distribution Obtained with options framework

The following figure shows the error compared to the target distribution. The horizontal axis corresponds to the number of iterations and the vertical axis corresponds to the error.

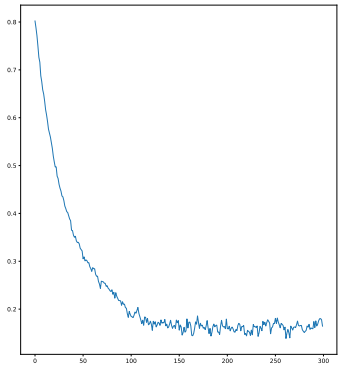

Fig. 5. Error with respect to the target distribution with the number of iterations

We further investigated that because of the stationary distribution of the markov chain, if the number of swarms are reduced to half the remaining swarm readjusts themselves. We test this particular property in the options framework for and show that swarm reconfiguration happens and the swarm readjusts to form the target shape when the number of agents reduces to half.This is shown in the following figure.

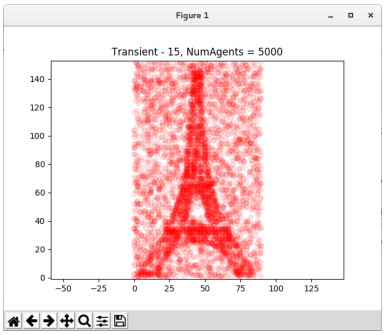

Fig. 6. Swarms forming the target distribution

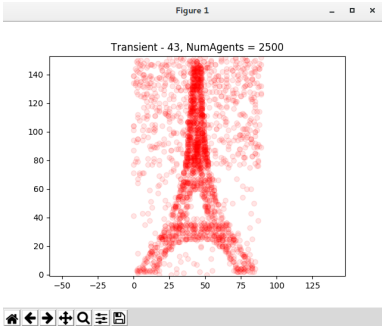

Fig. 7. Swarms reconfiguring when the number of particles reduces by half

V. Conclusion

In this paper we made an empirical study of swarm forming different shapes with a known target distribution. We also provided an extension in options framework for the swarms and provide the algorithm. Since this is an interesting problem with lots of application, we plan on to extend it further and use Hierarchical Reinforcement learning architecture. Since we were constrained by the time, we could only manage to extend it in terms of options, however we are still formalizing a way to learn to generate the matrix when the dynamics of the system is unknown. We also plan on extending this approach in terms of mean field team problems towards formation of the shapes.

References

[1] Bandyopadhyay, Saptarshi, Soon-Jo Chung, and Fred Y. Hadaegh. "Probabilistic and Distributed Control of a Large-Scale Swarm of Autonomous Agents." arXiv preprint arXiv:1403.4134 (2014).

[2] Acikmese, Behcet, and David S. Bayard. "Probabilistic swarm guidance for collaborative autonomous agents." American Control Conference (ACC), 2014. IEEE, 2014.

[3] Bacon, Pierre-Luc, Jean Harb, and Doina Precup. "The Option-Critic Architecture." AAAI. 2017.