Summary of Energy-Efficient 3D Vehicular Crowdsourcing For Disaster Response by Distributed Deep Reinforcement Learning

John Dewey, jcd18c
jcd18c@fsu.edu

1 INTRODUCTION

Natural disasters, such as earthquakes, hurricanes, and explosions, cause large amounts of damage, injuries, and casualties in short periods of time. Urgent situations require immediate action by rescue teams, as it is pivotal to do as much as possible in the "golden 72-hours" since after which survival rate falls to about 5-10%. Use of unmanned vehicles (UVs) can improve efforts in disaster response as their deployment flexibility and information collecting ability can reduce amount of work required from people, explore disaster areas that are impossible for humans to explore, and focus the efforts of humans on rescue. However, the amount of research in dynamic UB trajectory planning in extreme conditions is sparsely researched, and there are a few challenges that must be addressed to using UVs to explore disaster zones such as making UVs explore the complete disaster area, making UVs cooperate, and planning trajectories of UVs in disaster areas with unevenly distributed point-of-interests (PoIs). To address this, DRL-DISASTERVC(3D) is proposed.

2 RELATED WORK

2.1 Spatial Crowdsourcing (SC)

SC has been widely studied in both theoretical and various inductrial applications like web mapping services, ride-hailing services, and online search and recommendation systems. One of the key issues in SC is task assignment (TA) where workers are allocated to spatial tasks to maximize/minimize a total weighted value. Two categories of TA are online and offline scenarios. For online scenarios, Liu et al. proposed a threshold-based greedy algorithm in "Budget-aware online task assignment in spatial crowdsourcing." For offline scenarios, Li et al. proposed a 3D stable spatial matching solution in "," and Ni et al. proposed a game-theoretic approach to find an optimal worker-task routing path that considers task dependencies in "".

2.2 Deep Reinforcement Learning (DRL)

Reinforcement learning (RL) is widely used for sequential decision-making problems through iteratively interacting with a time-slotted environment, and are usually formulated by a Markov Decision Process (MDP). Actions are generated by a policy where a state transitions to its next state with a reward. DRL bridges RL and deep neural networks (DNNs) since DNNs allow the ability to learn intricate patterns and representations. Some representative DRL approaches include DQN, Rainbow, A3C, and DPPO; however, the state of the art approach is IMPALA - the core of DeepMind AlphaStar. IMPALA simultaneously increases speed and decreases instability of DRL training, and it is considered the starting point of DRL-DISASTERVC(3D).

Conference'17, July 2017, Washington, DC, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3 PROBLEM FORMULATION

3.1 System Model

The vehicular crowdsourcing (VC) task considers a set of UVs (drones and unmanned ground vehicles), $\mathcal{U}$, and a set of PoIs, $\mathcal{P}$. The UVs work to explore the PoIs in a 3D workzone that contains a set of obstacles to avoid while exploring. The task duration is fixed and divided into T equal timesteps of τ length, and each timestep is contains two parts, UV movement and data collection. The UV movement phase, a UV moves from its current position to a new position using an agle vector comprised of a polar and azimuthal angle and a moving distance (movement velocity is fixed). In the data collection part, a round robin sensing policy is used to collect data from a number of the nearest PoIs to a UV. Under the assumption that PoIs have multiple antennas using orthogonal frequencies, transmissions from PoIs will not interfere with eachother, so only the large scale pathloss effect between a PoI and UV is considered when measuring signal to noise ratio (SNR) and transmission rate. If the SNR is below a threshold, the data is considered too noisy and unusable. Therefore, the amount of data collected relies on data collection time, transmission rate, data dropout amount, and the number of PoIs serviced.

3.2 Problem Definition

Four metrics are used to define the UV navigation problem: data collection ratio, data dropout ratio, geographical fairness, and energy efficiency. The data collection ratio measures the average ratio of data collected from all UVs and the initial amount of data available data at all PoIs apon task completion, the data dropout rate measures data loss and the quality of the data collection due to impact of low SNR, geographical fairness uses Jain's fairness index to measure diversity and uniformity in the disaster workzone, and energy efficiency measures energy consumption during the task and combines the previous three metrics to achieve the goals of each simultaneously. This allows the energy efficiency metric to measure the overall performance of the task; therefore, the optimization problem can be viewed as maximizing energy efficiency. However, maximizing

4 SOLUTION

DRL-DISASTERVC(3D) is a heuristic DRL method that consists of a distributed, asynchronous DRL framework. The framefork is based on IMPALA and uses a repetetive experience relay for improved learning efficiency an attentive 3D convolutional neural network (CNN) with auxiliary pixel control for spatial exploration.

4.1 Distributed DRL Framework with Repetetive Experience Relay (RER)

A multi-actor-one-learner architecture is adapted from IMPALA where multiple actors (UVs) asynchronously work to generate MDP tuples by interacting with an environment. However, in IMPALA each MDP tuple is only used once, and the quality of learning from sampled experiences determines the accuracy and speed of training. Time and accuracy are essential in the domain of disaster recovery, so the team behind the paper introduces RER to enhance learning stability. There RER stores b_M batches of experiences and can traverse these experiences b_k times. After visiting a batch, the RER scrolls to a new batch. Everytime a batch is visited, a counter is updated, and when the counter reaches b_k, the batch is thrown away and replaced with a new one. One thing to note is that the dimmensions of the action space for multiple UVs expands tremendously compared to a single UV case, and this expansion leads to larger differences in local policies used by UVs. To stabilize this, a clipped target network is introduced. The target network of policies is periodically synchronized with the learner's policy network output. **Finish Clipped Target Network Part**

4.2 Attentive 3D CNN convolution with Pixel Control

A 3D CNN with multi-head-relational attention (MHRA) is used for spatial modeling, and auxiliary pixel control is added for spatial exploration.

4.3 Algorithm Design

There are two parts to the algorithm, multiple actors and a central learner.

5 EVALUATION

To evaluate the proposed model, the team behind the paper designed a simulator (DisasterSim) for VC in disaster response scenarios.

5.1 DisasterSim

DisasterSim was built using Unity 2018.3.14f1, Python 3.7.7m and Tensorboard 2.3.0. There are three layers to the simulation, the data layer, model layer, and visualization layer. The data layer contains scene configuration and model data...

5.2 Experiment

The team conducted an ablation study to measure the effectiveness of MHRA and PC and tested how hyperparameter tuning can affect performance.

5.3 Results

Results were good.

6 STRENGTHS

Paper very strong.

7 WEAKNESSES

Tough one here.

8 YOUR SOLUTION

I don't have a better solution.