Constrained Robot Path Planning in Wireless Communication

Project Plan

Anushka Vashishtha (3035299404)
FYP18049

Project Background

In the last years, Internet of Things (IoT) has taken the centre stage in the technology world by creating one of the fastest growing markets, as it has been predicted that there will be more than 30 billion connected devices by the end of 2020. In addition to that, IoT is already estimated to be generating 100s of trillion gigabytes of data per year and this figure is only increasing [1]. In the near future, almost every device will be connected to the internet, ranging from sensors, vehicles, wearable electronics to other embedded systems like refrigerators.

To prepare for the future, design engineers are working on finding efficient solutions in order to power as well as to communicate with these billions of devices, since providing sufficient energy to them consistently is quite a difficult task. Relying on traditional resources like batteries can meet the requirements but the downfall for them is that they must be bought, repaired and disposed sustainably [1]. Moreover, maintenance of such a battery becomes a challenge, when IoT devices work in inaccessible areas.

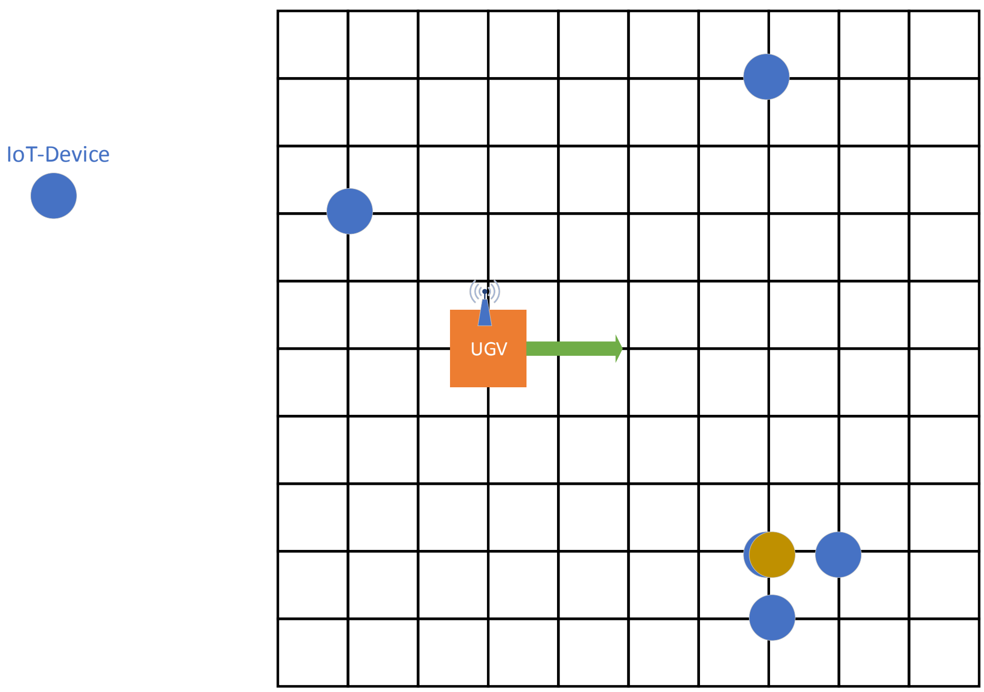
To solve this problem, an unmanned ground vehicle (UGV) is used to power (and communicate with) a cluster of IoT devices [2]. This will make traditional batteries an obsolete solution for powering the IoT devices and it will also resolve the difficulties involved in purchasing and maintenance. Therefore, rather than buying say 100 different batteries for 100 different IoT devices, there will now be just one UGV to power all of the devices and also collect data from them, if required.

Project Objective

From a Computer Science prospective, this problem can be abstracted to be a constraint based path planning problem in a graph. The unmanned ground vehicle will be capable of interacting with all the devices and charging all the devices at its location as well as the remaining devices present at different locations in a graph, all at once. To achieve this, I will be using different approaches such as Reinforcement Learning with Mixed Integer Non-Linear Programming (MINLP), Bandit Optimisation and Deep Reinforcement Learning with States encoding and Action embedding [3] [4]. After individually getting the results for the optimum paths from the various approaches mentioned above, I will systematically compare their performances to conclude which has a promising solution theoretically and if it is practically viable to apply any of the approaches in a real world environment.

It is the first time that this sort of approach is being used in interacting (charging/communicating) with IoT devices, therefore there is no prior work available in this arena.

Project Methodology



The abstraction of the problem gives the following model:

There is a set of IoT devices $\mathrm{D} \coloneqq \{\mathrm{d}_1, \cdots, \mathrm{d}_\mathrm{K}\}$ which are positioned on the vertices $\mathrm{V} \coloneqq \{\mathrm{v}_1, \cdots, \mathrm{v}_\mathrm{M}\}$ of an equal-distance-grid-shaped-graph representing the operation area. The UGV is located on one of the vertices v_U. In each time-step, the devices as well as the UGV can move one step along the grid. The UGV has to interact (i.e. provide power or communicate) with all devices at all times. The higher the distance between the UGV and a device, the more energy is needed for the UGV to interact with the device. The distance is determined using some distance measure (example Euclidian distance) on the positions of the UGV and the device on the graph. Also, if the UGV moves, a certain amount of energy is required.

To put this in a reinforcement learning setting, the statespace, the actions and the rewards are proposed to be defined in the following way:

Here, $v_{d_i}(t)$ is the position of the device i at time t in the graph. So, $v_{d_i}(t) = v_j$ represents that the device i at time t is at position of vertex j.

States S$:= \{(v_{d_1}(t) = v_i, \dots, v_{d_K}(t) = v_j, v_{UGV}(t) = v_p), \dots\}$

Actions A$:=$ $\{$"do not move UGV", "move up UGV, "move left UGV",$\dots\}$

Reward R(t) $:= -w_{movement} \begin{cases} 1 \; if \; UGV \; has \; moved \; so \; v_{UGV}(t-1) \neq v_{UGV}(t) \\ 0 \; otherwise \end{cases} - w_{interaction} \sum_{d \in D} f(distance(v_d(t), v_{UGV}(t)))$

Where $w_{movement}$ is a weight constant for weighting the importance of moving (or respectively the amount of energy needed for moving) and $w_{interaction}$ stands for the weight of the interactions. f is a strictly increasing function. The goal of the UGV is to maximize reward. The sum in the last term might also be replaced by the max over the values, whichever makes more sense from an electrical engineering perspective.

Note the minus signs: With positive weights, it is beneficial not to move and to be close to the devices.

For the case that the devices do not move there is one single optimal position in which the UGV should stay. The problem gets interesting because the devices move in an unknown manner. Therefore, the UGV always needs to readjust its position, maximizing its reward.

To solve this, different techniques will be used namely Q-Learning and MINLP.

Next, I will be using Deep Reinforcement Learning to approach the problem. Here, I will be encoding the states and inputting them into the neural net. Additionally, action embedding will play an important role. [3]

Lastly, the problem can be extended by removing the aspect of full information: What happens if the UGV does not have the information about the whereabouts of all devices (but just of some or none of them) and also no information about the size of the grid? The States, Actions and Rewards would stay the same, but the UGV would not have access to the whole information about the state.

This makes the problem resemble reality better, as it is difficult to track the devices' locations. Assuming the UGV only knows the locations of nearby devices, should it just stay at its current location and exploit the (relatively) high reward at the local optimum, or should it move around to explore a potentially better area with a global optimum?

Nonetheless, it also makes the problem more complicated to solve, so a version of the multi armed bandit problem will be applied. Therefore, for analyzing this case, heuristics to generate a feasible solution are used first. These heuristics are present in the set S $:= \{ S_1, S_{2,} \ldots, S_N\}$ where N can be say 5. These heuristics can be algorithms like MINLP, Q-Learning, simple Neural networks and their variants.

Now say there is a simulator that can calculate the reward R_g to the action determined by the algorithm S_g which the Bandit Optimizer chose [4].

Goal: best algorithm for the UGV = $\underset{S_g \in S}{\arg\max}\, E(\, R_g)$

Project Schedule and Milestones

September 30	Deliverable of Phase 1 • Project Plan • Project Website
October	Working into MATLAB, Python and TensorFlow Reading up on MINLP, Bandit Optimization and Deep Reinforcement Learning
Nov - Dec	Development of demo application • Creating simulated environment • Applying MINLP and Deep Reinforcement Learning
Jan	Deliverable of Phase 1 • Demo Application • Interim Report
Dec - Feb	Applying Bandit Optimization to partially known state space
Mar - Apr	Comparing the results obtained from the different approaches
April	Deliverable of Phase 3 • Finalized Implementation • Finalized Report

Bibliography

[1] A. Karmakar and R. Desai, "Powering IoT Devices Through Energy Harvesting: Present And Future Trends," 5 6 2017. [Online]. Available: https://www.ecnmag.com/article/2017/06/powering-iot-devices-through-energy-harvesting-present-and-future-trends.

[2] S. Wang, M. Xia and Y. C. Wu, "Backscatter data collection with unmanned ground vehicle: Mobility management and Power allocation," *IEEE transcations on wireless communications,* p. 30, 2018.

[3] E. Howley, M. Schukat and S. S. Mousavia, "Deep Reinforcement Learning: An Overview," in *SAI Intelligent Systems Conference*, 2016.

[4] D. Precup and V. Kuleshov, "Algorithms for the multi-armed bandit problem," *Journal of Machine Learning Research*, vol. 1, pp. 1-48, 2000.