**Project 1 - Brainstorming Report**

1. What would you define as an AI search problem in wilderness search and rescue? There may be more than one possible application in WSAR, list as many as you can think of AND the assumptions about that problem specification. Think of each application as a software agent.

* Let us define a few terms to establish the context of the *AI search* problem.
  1. Search and rescue (SAR) is the search for and provision of aid to people who are in distress or imminent danger.
  2. Wilderness Search and Rescue (WiSAR) refers to SAR operations carried out in non-urban/natural environments like forests.
* Looking at the above definitions, the scope of the task for a software agent might be too broad for WiSAR. Therefore, we refine the search problem as: the search for and provision of aid to people who are in distress or danger, with an emphasis on information gathering and sharing.
* Thus, our software agent would primarily be involved with gathering information and working in conjunction with *external agencies* in a *cooperative* manner.
* Application could be:
  1. An embodied software agent, paired with a robot body like a UAV, UGV, or ROV. Here, the agent is involved in a multi-agent environment with the task of exploration/search of points of interest (POI).
  2. A software agent running on a data center, whose search task is to sieve through information – high resolution photographs, video – as quickly as possible and relay new information based on interest markers found within the data.
* For both the agents, time is a key metric – we need to transform information to knowledge as swiftly as possible.

1. For each of the AI search problems in #1, describe the agent in terms of the P and E in PEAS.
   * 1. P: Generate as many performance metrics as you can think of, then select two that are most promising.
     2. E: describe the domain in terms of environmental dimensions: Observable, Deterministic, Episodic, Static, Discrete, Single-agent? and give your justification in two to three complete sentences

* Application #1:

|  |  |  |
| --- | --- | --- |
| Performance measure (**P**): | | |
| Metric #1 | Area covered | Measure the area covered with respect to total search area as a function of time |
| Metric #2 | POIs explored | Measure the POIs explored |
| Metric #3 | Subject detection | If the subject is found or not; optimal performance measure |
| Environment (**E**): | | |
| Observable | | Partial; even though we might have a map of the area, it may or may not be up-to date. |
| Deterministic | | No; because the environment is partially observable |
| Episodic | | No; the environment is sequential. For e.g. The agent might pursue a different search path if a trail (footsteps) is found. |
| Static | | No; the environment is dynamic. For e.g. there could be a change in weather, altering the known environment. |
| Discrete | | No; the environment is continuous; assuming we map the environment as a set of continuously varying geographic coordinates. |
| Single-agent | | No; we have assumed the existence of external agencies of cooperative nature. E.g. SAR volunteers. |

Of the performance measures listed, metric #1 and #2 seem to be more appropriate in guiding our agent on a converging state.

* Application #2

|  |  |  |
| --- | --- | --- |
| Performance measure (**P**): | | |
| Metric #1 | Data investigated | Measure of the data investigated. E.g. how many photographs have been analyzed out of the total |
| Metric #2 | POIs detected | Measure the data that has been marked important. E.g. the photographic markers placed on ‘important’ images |
| Metric #3 | Subject detection | If the subject is found or not; optimal performance measure |
| Environment (**E**): | | |
| Observable | | Yes; we have all the data with us for analysis. |
| Deterministic | | Yes; the data is static and not changing with time. |
| Episodic | | No; the environment is sequential. For e.g. The agent might look through a different sequence of data if the current photograph shows POIs. |
| Static | | Yes; the environment is static. The data we have for analysis does not change with time. |
| Discrete | | Yes; the data we have is fixed, i.e. we can count them quantitatively in a fixed precision. |
| Single-agent | | Yes; we have assumed the agent works on it’s own. |

Of the performance metrics listed, metric #1 and #2 seem more promising.

1. For each AI search problem in #1, discuss why or why not each of these categories of search is appropriate or not appropriate. You will need to refer to the table and discussion in #2 as well as any other relevant factors that you think are important. For each category of search that is appropriate, suggest at least one specific technique and describe why it is a good choice. You will want to look through the book to see if any of the techniques we didn’t cover are applicable before.

* Application #1:
  + Blind search techniques like BFS is not applicable since the environment is partially observable.
  + Heuristic search techniques like A\* is not applicable because the environment is partially observable.
  + Adversarial search techniques like MINIMAX is not applicable since there is no definable adversary.
  + It would be overkill to define the continuous, partially observable environment as a Constraint Satisfaction Problem.
  + Therefore, we go with Local Search techniques like Hill Climbing.
    - Before we dive into how the search can be solved by Hill Climbing method, let us model our environment as a probability distribution measure of various heuristics: Subject profile, weather profile, equipment profile, subject experience profile, terrain and hazards profile, history of incidents in this area etc. We could get the following generated map –

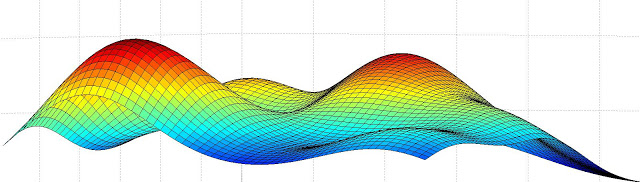


Figure 1. 3D probability distribution surface

* + - We can apply the Hill-Climbing search as follows: We start from the LKP (last know position) point of the surface and then check its. As soon as a neighbor with the same or better value is found, we "climb" to that point. The process is repeated until we reach a point (hilltop) where all neighbors have smaller values. As we "climb" and check neighbors, we mark all the points we visited along the way. And when we check neighbors, we only check points we have not visited before. Once we find a “hilltop”, we can start from another unvisited point on the surface and do another Local Hill Climbing.
  + Another way to tackle the problem is by generating *belief states* and then solving the problem using the And-Or Graph Search algorithm.
* Application #2:
  + Blind Search algorithms like BFS may not be the best strategy, given that time is limited. Also, we would fail to apply the various heuristics we discussed above.
  + Heuristic Search algorithms like A\* and Greedy BFS seem to fit the bill in terms of the environment, however, the search problem here is to identify states which are of interest. We are not concerned with the steps.
  + Local search algorithms could be beneficial in that we could explore high “score” states.
  + Adversarial search techniques like MINIMAX is not applicable since there is no definable adversary.
  + To apply a local search technique like AO\* algorithm, we could generate the search tree as follows:
    - Divide the search tree into areas that correspond to high probability regions. See figure 1.
    - Generate the states of children of these subtrees by clustering all the data pertaining these regions into that state in a sequential manner.
    - We start our search by looking at the sub-tree of highest probability, exploring the states and identifying POIs, if any. For e.g. Man-made structures. We mark these states in memory.
    - If a state has many POIs, we explore its children sequentially. Otherwise we mark that state as *unsolvable* and move on to other subtrees.
    - We terminate the search if the subject is detected, or else we return the states we stored above.

1. Where else could AI (beyond AI search) be useful?

* In applying our AI search technique for application #1, we generate a 3D probability map. To obtain such a map we could use a software agent to do the *mapping* for us. Another area to apply AI is *planning*. After constructing our map, we could apply AI to obtain the best path plan factoring in constraints.

References

* *http://lannyland.blogspot.com/search/label/My%20Research,* visited on Feb 5, 2018.
* *Artificial Intelligence: A Modern Approach,* 3rd Edition, S. Russell and P. Norvig