

# Summary of DRA\* Algorithm

Saksham Goel

September 22, 2018

## Abstract

DRA\* also referred to as Dynamic Repairing A\* algorithm is a proposed algorithm to tackle the problems of re-planning. The author suggest that in the context of re-planning, DRA\* algorithm performs better with the situation of change in goal state or action weights by using the already computed states as compared to A\* which would use a strategy of computing the answer from scratch. This assignment is used to summarize the paper and answer three critical questions.

## 1 The authors claim this algorithm is useful in “dynamic terrain”. What is a specific examples of a “dynamic terrain” where this algorithm would be useful?

Finding directions or shortest paths between locations on maps in a real world scenario is a dynamic terrain environment. To explain why these situations are dynamic terrain, consider the simple example of traffic accumulating on a particular path because of some unfortunate accident or spill or certain unforeseen circumstances. Accident can lead to road closures and change in certain routes, which is like changing the action weights or weights of paths (time) between two locations. For e.g. if an accident occurred on a road and there is too much traffic accumulate there because of closure of 2 out of 4 lanes then the approximate time taken to cover the distance between the start and the end of that road increases.

I think this is where the DRA\* algorithm would be really useful. As stated in the conclusion section, DRA\* algorithm performs very well when the action weights changes as compared to A\*, this is the perfect dynamic terrain environment where a small change in action weight (time of travel along a certain path) due to an accident should not lead to reevaluating the whole shortest path from scratch but use some computations from earlier to achieve the desired effect and compute the new shortest path (some detour from the current location without any traffic). I think Google Maps, Apple Maps etc can use or are already using this kind of search to save a lot of time.

Other examples include a hedge maze where after certain point in time the paths gets closed adding small patches of hedges in between paths (like in Harry Potter) or another environment like Mars curiosity rover which may be looking

for something on Mars, however as time passes and because of some unforeseen changes the possible location of the object of interest may have shifted then it should not do all the computations again, however just make sure to account for the minor changes in the location of the object of interest and take the next steps appropriately.

## **2 What is an example of a “dynamic terrain” where the proposed algorithm might not be useful?**

Dynamic terrain like multiple robots mining for some mineral in a particular cave can be an example where the proposed algorithm might not work, because if the mineral of interest (wants to mine some other mineral) changes then it is better reevaluate the whole land and start fresh from some piece of land with higher probability of containing the mineral rather than backtracking from the earlier mining site and then trying to mine till reach the better location and then start mining from there.

## **3 Does the proposed algorithm always outperform the other “state-of-the-art” algorithms?**

Considering that the question referred to the algorithms - (Stentz et al., 1995; Koenig and Likhachev, 2002; Likhachev et al., 2003) (Hansen and Zhou, 2007), (Koenig et al., 2004; Van Den Berg et al., 2006; Koenig and Likhachev, 2006) as the state of the art algorithms dealing with re-planning, then I think the answer to the question cannot be answered, because in the paper the author does not compare the performance of DRA\* algorithm with respect to algorithms mentioned in the previous papers. According to the author the DRA\* algorithm cannot be compared with the others because the other algorithms are utilized in specific settings which DRA\* does not examine.

Considering that the other state of the art algorithms refer to searching algorithms like A\*, I think according to their evaluation section, Author mentions that the DRA\* algorithm is better than A\*, however only certain given conditions. The performance of DRA\* with respect to A\* greatly depends on the given graph structure (like branching factor) and also the amount of change in the given re-plan. As mentioned in the results section, the optimal performance of the DRA\* algorithm is much better than A\* only if the original plan has not been executed more than 50% and the change in goal set is not more than 20 - 50%. So I would not necessarily say that it outperforms the other state of the art algorithm however it certainly performs better than some of them in certain conditions.