AUTOMATIC PLANNING - REPORT - LAB 2

Manu S. Joseph(mansj125)

Lab 2.1. Emergency Services Logistics, Carriers

Question 1)

The domain was modified to include carriers. New types called *carrier* and *num* were introduced for carriers and numerical values. Numerical values were required to keep track of the free space inside carriers. Three new predicate were added to the domain called,

- *carrier_crate* - used to associate a crate to a carrier when the crate is loaded in the carrier.
- *carrier_freespace* - used to keep track of the free space in the carrier. (i.e. the number of crates that can be loaded into the carrier).
- *next_num* - used to give an order to the number objects

Three new action were added to the domain called,

- *load_crate_on_carrier* - loads crates onto a carrier. This associates the crate with the carrier using the *carrier_crate* predicate and decrements the free space in the carrier. Preconditions makes sure that helicopter and carrier are in the same location, and if the carrier has some free space left. Preconditions for this action also checks if the helicopter already has a crate with it so that it can be placed into the carrier (This can only be done using the pickup action that was introduced in the last lab.)
- *fly_carrier* - This actions flys a carrier from one location to another. Preconditions make sure that the carrier and helicopter are in the same location and also if the helicopter is free(i.e it is not carry any carriers or crates).
- *take_crate_from_carrier* - This action takes a crate from a carrier and associates it with the helicopter allowing the helicopter to give the crate to some person using the *deliver* action. This action also removes the association of the crate with the carrier and increments the carriers free space. Preconditions of this action makes sure that the carrier and helicopter are in the same location and if the helicopter is free.

The domain definition file is named - *domain.pddl* and comments are added to the file for more information.

Question 2)

Three problem instance were generated after modifying the problem generator.

- *Small sized problem* - 1 UAV, 1 carrier(free space - 2), 1 location, 2 people, 2 crates and 2 goals. *(File name - small_uav_problem_u1_r1_l1_p2_c2_g2_ct2.pddl)*
- *Medium sized problem* - 1 UAV, 1 carrier(free space - 4), 4 location, 6 people, 6 crates and 6 goals. *(File name - uav_problem_u1_r1_l4_p6_c6_g6_ct2.pddl)*
- *Large sized problem* - 1 UAV, 1 carrier(free space - 4), 4 location, 10 people, 10 crates and 10 goals. *(File name - large_uav_problem_u1_r1_l4_p10_c10_g10_ct2.pddl)*

The medium sized problem was selected so that it can be solved by *IPP* in around 1 minute. The large sized problem is just a bigger version of that problem. The small sized problem created to check if the planners would be able to find solution with carrier when the search state space is very small.

Question 3)

The problems generated was run using the planners that was introduced in lab 1.

For the small problem the *Cerberus* planner found a plan using carrier but the plan was not optimal. It found a plan that loads 1 crate into a carrier and fly the carrier to the destination, unloads and delivers it and repeats this again. This is a bad plan. No other planners found a solution with carriers for the small problem.

For the medium and large problems only the *Madagascar* planner found a solution with carriers even then the solution was pretty bad.

The outputs of running these planners on the problems are present in the *Output_lab2.1* folder.

Lab 2.2. Emergency Services Logistics, Action Costs

Action costs were incorporated into the domain and problem file. A constant cost of 10 was given to the actions - *pickup_crate*, *deliver*, *load_crate_on_carrier* and *take_crate_from_carrier*. The action cost for *fly_to* and *fly_carrier* was generated using the flight_cost function in the generator file.

Question 9)

The problems modified to include action costs were run using the same planners as before.

For the small problem - *Cerberus*, *Lama* and *Jasper* planners were able to find the optimal solution using carriers as their second plan which saw helicopters using carriers as it was better to use carriers according to the distance. The Madagascar planner failed to find solutions using carriers this might be because of the fact that Madagascar does not consider action costs.

For the medium size problem and the large size problem - It could be seen that all planner were eventually able to generate better solutions using carriers after the addition of action costs. The first solution generated by the planners generally did not use carriers or used carriers in a bad way but better plans were subsequently generated by the planners that used carriers more efficiently and had less action costs.

The plans that were generated after adding action cost are considerably better than that which were generated earlier without action cost. Even if the initial plans are inefficient subsequent plan were better and efficient in which helicopters makes more use of carriers to deliver crates. On the other hand without action cost the planner would rarely create plans with carriers and even if they are created they would inefficiently or unnecessarily use carriers.

The outputs of running these problems with the planners can be found in the folder *Output_lab2.2*

Lab 2.3. Emergency Services Logistics, Optimal Planners

The problems generated were tested using the optimal planners slemax, spams, BJOLP and symba1.

For the medium sized problem,

All the optimal planners were able to find the optimal solution with the plan cost as 865. The plan generated by each of the planner varied in the order in which things were done but had the same plan cost. The plans were generated by the planners in the following time,

- *BJOLP* - took 8 seconds to generate optimal plan.
- s*elmax* - took 63 seconds to generate optimal plan.
- s*pmas* - took 12 seconds to generate optimal plan.
- s*ymba1* - took 3.5 seconds to generate optimal plan.

But plans with the same cost of 865 were also found the following seq-sat planners,

- *cerberus* - generated the optimal solution in 12.475 seconds and as the 6th solution.
- *jasper* - generated the optimal solution in 0.02 seconds and as the 2nd solution.
- *lama* - generated the optimal solution in 60 seconds and as the 6th solution.
- *madagascar* - did not generate an optimal solution, but the solution generated used carriers but the actions done was unnecessary.

For the larger problem,

The optimal planners failed to find solution even after searching for around 5 minutes. At the same time the seq-sat planners found some non optimal solutions for the problems quickly and then also found some improved plan that are significantly better than the first solution after some time. This the data that is got from running the seq-sat planners for 1 minute,

- c*erberus* - Found the fist plan with the plan cost of 1998 in 0.05 seconds. The best solution found within 1 minutes has the cost of 1260 and was found in the 10.7th second.
- *jasper* - Found the fist plan with the plan cost of 1998 in 1 second. The best solution found within 1 minutes has the cost of 1260 and was found in the 54th second.
- *lama* - Found the fist plan with the plan cost of 1998 in 0.02 second. The best solution found within 1 minutes has the cost of 1260 and was found in the 50.73th second.
- m*adagascar* - Found plan without use of carriers

The outputs of the seq-sat planners for the problems are found in the folder - *Output_lab2.2* and for the optimal planners are found in the folder - *Output_lab2.3.*

It can be understood from this data that the satisficing planners generate plans that are not optimal much more quickly than optimal planner which search until the optimal plan is found. The optimal planner might be able to generate the optimal plans a bit faster than the satisficing planners but it might not be easy to always find the optimal plan, this is clear form the data that is gathered from running the larger problems.

For large problems optimal planners would take a very long time to generate a plan this would not work out in all domain. Optimal planners might not work out for the domain in this lab which is emergency service logistic. Time is very critical when it comes to emergency service. Taking too much time to generate optimal plans would severely endanger many people. So generating sufficiently good plans quickly using seq-sat planner and selecting the plan with least cost generated in a specified amount of time would be best in this domain.