A Domain-dependent solver for Tidybot

Supervisor: Álvaro Torralba, alto@cs.aau.dk

August 24, 2020

Tidybot is a domain used to evaluate domain-independent planning systems, and was introduced in the 2011 International Planning Competition with the following description:

> The Tidybot domain models a household cleaning task, in which one or more robots must pick up a set of objects and put them into goal locations. The world is structured as a 2D grid, divided into navigable locations and surfaces on which objects may lie. Robots have a gripper, which moves relative to the robot, up to some maximum radius. Existing objects block the gripper, so that it may be necessary to move one object out of the way to put another one down. Robots can carry one object at a time in the gripper, but may also make use of a cart, that can hold multiple objects. The instance generator creates worlds that contain rectangular surfaces ("tables"), as well as U-shaped enclosures ("cupboards"), which are the goal locations of objects.
> In many real-world problems, the difficulty is due to the large state space and number of objects, rather than due to complex, "puzzle-like" combinatorial constraints. Humans are able to quickly find feasible solutions in such domains, because they seem to be able to decompose the problem into separate parts and make use of the geometrical structure. This domain is thus intended to exercise the ability of planners to find and exploit structure in large but mostly unconstrained problems.
> Optimal reasoning in such problems is challenging for humans as well, and a secondary motivation for the domain is to test the ability to do optimal reasoning in geometrically structured worlds. The presence of the carts adds another combinatorial decision: it might be worthwhile to spend some time fetching the cart to avoid later having to go back and forth with each object.

The purpose of this project is to develop a controller for a household robot, taking as a basis the model of the Tidybot domain. The input to the controller is a description of the current state of the world, including:

- The dimensions of the 2D grid, and the position of all the relevant obstacles
- The current position of the robots' base and gripper
- The current and target positions of the objects to be moved
- The position of the carts'

The controller must return a sequence of actions that achieves the goal, while minimizing the cost. The results should be compared against domain-independent planners.

Theoretical Approach: Develop an algorithm with theoretical guarantees on solution quality, i.e., optimal solutions or bounded-suboptimal solutions. Identify properties of the problem, and develop algorithms that exploit them while looking for the best solution. For example, under which conditions will be useful to use the cart to transport a set of objects? If one can identify sufficient conditions under which one can prove that using the cart is (or is not) a good strategy, that may speed-up the search for a solution.

Practical Approach: Develop an algorithm that achieves as good solutions as possible in as little time as possible. Analyze the trade-off between solution quality and efficiency, and how different parameters of the algorithm influence them.