HTN IPC-2020 Domain: Robot

Ron Alford
The MITRE Corporation
ralford@mitre.org

Abstract

The Robot problems from the 2020 HTN International Planning Competition were adapted from the HTN translation papers by Alford et al.. The domain encodes an office delivery problem, and enforces a strategy of only picking up packages that are not in their goal location, and only placing packages in their final location. Package order and navigation are left up to the choices of the planner.

The Robot domain, modeled after the Robot Navigation domain (Kabanza, Barbeau, and St-Denis 1997), was used to evaluate the translation of totally-ordered HTN problems to PDDL (Alford, Kuter, and Nau 2009) and updated in further work translating partially-ordered HTNs to PDDL (Alford et al. 2016). This domain was originally designed to showcase the effectiveness of partial HTN knowledge in guiding planner search. The domains in the above papers encoded a method for picking up objects in their initial locations and placing them in their goal location. Navigation actions were allowed to be inserted by the planner outside of the HTN structure, similar to HTN task insertion planning (Geier and Bercher).

In the IPC, the additional methods were introduce to place the navigation actions in the hierarchy, giving the domain traditional HTN semantics. The new domain contains 4 operators and 11 methods implementing 6 abstract tasks. The problems are tail recursive with a max progression bound of 2, which means any HTN progression of the initial task network results in a task network with at most two tasks (Alford, Bercher, and Aha 2015).

The top level task, `achieve-goals`, has four methods:

- `achieve-goals-move`, `achieve-goals-open`: These change rooms and open door, respectively, and then recurse back to `achieve-goals`
- `achieve-goals-pickup` has a precondition that there is an object in the current room and its not in its goal location. Its subtasks pick up the object, and recurse to the `release` task.
- `finished` has no precondition or subtasks.

The `release` task has three methods:

- `release-open` and `release-move` which call the open and move operators respectively, then recurse back to `release`.
- `release-putdown_abstract` has a precondition that the robot is holding a package while in the package's goal location. It has subtasks to release the package and recurse back to the top level task `achieve-goals`.

The problem generator[1] for the Robot domain takes the number of packages and rooms as a parameter. It generates a problem with the rooms connected in a random acyclic graph, the packages uniformly distributed among the rooms, and the doors closed or open with even chance. The problem generator outputs in PDDL format, which had to be hand adapted for the HDDL format of the HTN IPC (Höller et al. 2020). The IPC contained 30 problems with room counts between 1 and 300, and package counts between 1 and 150.

References

[Alford, Bercher, and Aha 2015] Alford, R.; Bercher, P.; and Aha, D. 2015. Tight bounds for HTN planning. In *ICAPS*.

[Alford et al. 2016] Alford, R.; Behnke, G.; Höller, D.; Bercher, P.; Biundo, S.; and Aha, D. 2016. Bound to plan: Exploiting classical heuristics via automatic translations of tail-recursive HTN problems. In *ICAPS*.

[Alford, Kuter, and Nau 2009] Alford, R.; Kuter, U.; and Nau, D. S. 2009. Translating HTNs to PDDL: A small amount of domain knowledge can go a long way. In *IJCAI*.

[Geier and Bercher] Geier, T., and Bercher, P. On the decidability of HTN planning with task insertion. In *IJCAI*.

[Höller et al. 2020] Höller, D.; Behnke, G.; Bercher, P.; Biundo, S.; Fiorino, H.; Pellier, D.; and Alford, R. 2020. HDDL: An extension to PDDL for expressing hierarchical planning problems. In *AAAI*.

[Kabanza, Barbeau, and St-Denis 1997] Kabanza, F.; Barbeau, M.; and St-Denis, R. 1997. Planning control rules for reactive agents. *Artificial Intelligence* 95(1):67–113.

[1] https://github.com/ronwalf/HTN-Translation/tree/master/examples/robot