Domain Form

(1) Name and email address of the contact person

Hootan Nakhost
nakhost@ualberta.ca

(2) Brief description of domain and problems.

NoMystery is a transportation domain designed to study resource-constrained planning [1,2]. In this domain, a truck moves in a weighted graph; a set of packages must be transported between nodes; actions move along edges, and load/unload packages; each move consumes the edge weight in fuel. In brief, NoMystery is a straightforward problem similar to the ones contained in many IPC benchmarks. Its key feature is that it comes with a domain-specific optimal solver allowing to control the constrainedness of the resources.

The generator first creates a random connected undirected graph with *n* nodes, and it adds *k* packages with random origins and destinations. The edge weights are uniformly drawn between *1* and an integer *W*. The optimal solver computes the minimum required amount of fuel *M*, and the initial fuel supply is set to $\lfloor C \times M \rfloor$, where $C \geq 1$ is a (float) input parameter of the generator. The parameter *C* denotes the ratio between the available fuel vs. the minimum amount required. The problem becomes more constrained when *C* approaches 1.

(3) Comments about why the domain is particularly interesting for the deterministic track

A ubiquitous feature of planning problems is the need to economize limited resources such as fuel or money. While heuristic search, mostly based on relaxation heuristics, is currently the superior method for most varieties of planning, its ability to solve critically resource-constrained problems is limited: relaxation heuristics basically ignore the resource consumption by actions. NoMystery generator is a nice test domain to study the behavior of planning algorithms in planning problems with resources.

(4) Comments about the ability to easily scale problem difficulty

There are two ways to scale problem difficulty in NoMystery: increasing the number of packages and locations, and decreasing C. The generator is very fast up to 18 locations and 18 packages.

Two types of encoding can be used for the problems: Hard and Hard-cost. The only difference is in the action costs. While all actions have a unit cost in Hard encoding, in Hard-cost version action costs are equal to the amount of fuel consumed by the action. Therefore, the total cost of a plan for Hard-cost encoding equals the amount of fuel consumed in the plan.

In Hard encoding, problems with 12 locations and 12 packages become quite challenging for state of the art planners when C is close 1 [1].

Hard-cost encoding makes the problems easier for the current planners. The reason is that the heuristic functions that consider costs are not any more completely ignorant to the resource consumption of actions. However, this type of encoding is not always feasible for resource planning; there might be several resources and the cost of the plan might be different from the amount of the resource consumption. However, our initial experiments show problems in this encoding with 12 locations and 15 packages become challenging for current planners when C is close to 1.

(5) Running the generator

To find out about the inputs of the generator run it with a random input.

Resources

[1] Nakhost, H.; Hoffmann, J.; and Müler, M. 2010. Improving local search for resource-constrained planning. Technical Report TR 10-02, Dept. of Computing Science, University of Alberta, Edmonton,
Alberta, Canada.

[2] Hoffmann, J.; Kautz, H.; Gomes, C.; and Selman, B. 2007. SAT encodings of state-space reachability problems in numeric domains. In Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI-07), 1918–1923