Discovering Generalized Landmarks: Capturing Abstract Relations in Real-World Planning

Issa Hanou[1], Sebastijan Dumančić[1], and Mathijs de Weerdt[1]

Delft University of Technology, The Netherlands
i.k.hanou@tudelft.nl

In AI planning, we want to find a plan by selecting the next action based on the current state and the goal to be achieved. Solvers for these problems use model-based approaches independent of the domain or specific problem instance. This creates a very general framework, allowing us to solve a wide range of problem types. AI planners have benefited from a range of heuristics, for example, by reaching landmarks: properties of states that you must reach on your path to reach the goal [4]. Landmarks have resulted in significant solving speed up, as the search through the state space can be prioritized toward a 'subgoal' on the way to the final goal. However, these landmarks have to be extracted for every problem instance before they can be used, and they are limited to the existing properties of the domain. Therefore, we propose a new type of landmark that we call *generalized landmarks*. These are computed for an entire domain of problem instances and consist of first-order functions so they can be more expressive.

The problem domain in AI planning consists of *predicates* over objects [3]. For each domain, we can have a set of problem instances with a different number of objects. For example, consider a package delivery problem in a grid world: there are locations, packages, and trucks, and the predicates could be `at` (location of packages and trucks), `empty`, and `carrying` (to specify the transportation of packages by trucks), and we define a goal location for all the packages. A traditional landmark might say that before we can deliver `package1` to `location3`, `truck2` must carry `package1`, and before that, `truck2` must be empty. This is quite limited and not very informative, so our new *generalized landmark* could simply say that any package that has to be delivered must first be carried by some truck; and prior, the truck must be at the same location as the package (without saying which location, which truck, or which package, specifically).

To define generalized landmarks that capture the above example, we no longer focus on the predicates of a domain but use a set of descriptive domain *features*, which are first-order functions over the domain predicates that can be either numerical or Boolean [1]. These features can capture more general properties of states, like a truck and a package being at the same location. Moreover, we introduce a *compound feature*, which is a conjunction of features that together describe a state. We can now define generalized landmarks as compound features that identify a state that must be reached to achieve the goal.

We want generalized landmarks to be more informative than traditional landmarks, and we propose a new method to discover them from previous plans. Since the plans already contain information on how to achieve this goal, we do not have to consider the complete state space of a domain but can limit our discovery process to the state space constructed by the state trajectories of the plans. So, given a small set of problem instances and several plans, we first draw up the state space of the state trajectories. Then, we compute a set of features describing the state trajectories, of which we select the compound feature that identifies some state in each trajectory. The landmark is then a compound feature that defines the earliest next state across all trajectories. This process can then iteratively detect a chain of landmarks.

Besides generalized landmarks being independent of the problem instance, they are also more powerful in capturing repetitive patterns in a plan. We detect the recurrence of already detected landmarks to identify loops in the landmark chain, thereby creating a graph of landmarks. Moreover, we can also detect the optional condition when the loop is possible. For example, consider the landmark graph in Figure 1 and the package delivery problem. Suppose we have two packages that have to be delivered at the same target location, then first we need to go to the package (landmark `PackageLocation`), then we need to pick it up (landmark `CarryingPackage`), next, we have to reach the dropoff location (landmark `GoalLocation`), and now we can deliver the first package. For package 2, we have to achieve the same steps, and then finally, we can deliver all packages (landmark `DeliverAllPackages`). The loop avoids the repetition of some landmarks, while being able to capture the landmarks for instances with one or more packages. The edge that creates the loop is thus not a must-take transition, and the condition specifies when you can take this transition.

We implemented the discovery in an Answer Set Programming model [2], which enables us to quickly find the compound feature that constitutes a landmark. We only need a small set of plans as input to find a landmark graph representative of larger instances in the same domain. We show the landmark graph can be used in planning by creating a new heuristic that counts the landmarks that have been achieved so far and gives higher priority to exploring states that have the most achieved landmarks, also respecting their orders and loop conditions. This method shows that generalized landmarks can indeed be used to find plans in new instances, but the planning process can still be optimized.

In conclusion, we propose a framework for discovering generalized landmarks and using them in a planner. Generalized landmarks generalize across instances in a domain, also scaling to larger problem instances than the training set. Furthermore, they capture symmetric and repetitive properties of plans and illustrate the necessary conditions for using repetitive plan fragments. Finally, they give an abstract representation of plans for a domain, allowing us to explain the steps for solving problems in this domain.

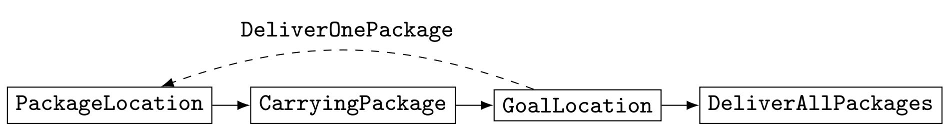

Fig. 1. Landmark Graph for Package Delivery Problem.

References

[1] Blai Bonet and Hector Geffner. "Features, Projections, and Representation Change for Generalized Planning". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. June 2018. URL: https://www.ijcai.org/proceedings/2018/649.

[2] Martin Gebser et al. *Answer Set Solving in Practice*. Morgan and Claypool Publishers, Dec. 2012. DOI: 10.2200/S00457ED1V01Y201211AIM019.

[3] Hector Geffner and Blai Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. 1st ed. Springer Cham, 2013. URL: https://link.springer.com/book/10.1007/978-3-031-01564-9.

[4] Julie Porteous, Laura Sebastia, and Jorg Hoffmann. "On the extraction, ordering, and usage of landmarks in planning". In: *6th European Conference on Planning*. 2001. URL: https://strathprints.strath.ac.uk/1874/.