# IPC Learning Track: Novelty-Based Generalized Planning

**Chao Lei, Nir Lipovetzky, Krista A. Ehinger**
School of Computing and Information Systems, The University of Melbourne, Australia
clei1@student.unimelb.edu.au, {nir.lipovetzky, kris.ehinger}@unimelb.edu.au

**Abstract**

It has been shown recently that successful techniques in classical planning, such as goal-oriented heuristics and landmarks, can improve the ability to compute planning programs for generalized planning (GP) problems. Besides fact landmarks, other ideas in classical planning have not been introduced to generalized planning, such as *novelty-based search*. In this paper, we present novelty-based generalized planning solvers, which prune a newly generated planning program if its most frequent action repetition is greater than a given bound $v$, implemented by novelty-based *Progressive Generalized Planning* PGP($v$). Besides, we introduce new structural program restrictions to scale up the search.

## Generalized Planning as Heuristics Search

Recently, Segovia-Aguas, Jiménez, and Jonsson (2019) proposed a PSPACE-complete formalism for GP problems whose solutions are *planning programs*, where a sequence of program instructions are nested with `goto` instructions such that the program can execute looping and branching structures. Candidate instructions are programmed in sequence, one line a time, while assessing whether the planning instances are solvable given a maximum number of program lines. The main algorithm that follows the heuristic search paradigm to search in the space of programs is known as Best-First Generalized Planning (BFGP) (Segovia-Aguas, Jiménez, and Jonsson 2021), where variable pointers and higher level state features allow programs to solve instances with different variables. To further scale up search efficiency, Segovia-Aguas et al. (2022) introduced a progressive search to avoid over-evaluating whether each subprogram in the search is a solution for the entire set of instances. Progressive GP (PGP) starts a Best First Search (BFS) with an empty program $\Pi$ of at most $n$ program lines, and the first instance as the only *active instance* (Segovia-Aguas et al. 2022). Search nodes are generated by programming up to $n$ instructions while pruning nodes recognized as dead-ends. The underlying BFS expands the best $\Pi$ in the *open* list according to its evaluation functions. PGP returns $\Pi$ as a verified solution if $\Pi$ solves all active instances and has been validated as a solution in the remaining non-active instances. If the validation fails, one of the non-active instances is added to the active instances, and the open list is reevaluated. A GP problem is unsolvable if active instances include all instances but no solution is found. If all instances are active when the search starts, then PGP is equivalent to BFS.

## Action Novelty in Planning Programs

The notion of *novelty* was first introduced by Lipovetzky and Geffner (2012) in classical planning to assess how novel a state $s$ is with respect to a given context $C$, defined as the states already visited by the search strategy. In classical planning, novelty is defined in terms of the predicates of a state, while, in GP, each search state is defined by the actions assigned to each line in a planning program. As a result, we define the *novelty rank* of an action $a^*$ where $a^*$ is a action schema or RAM action (Segovia-Aguas, Jiménez, and Jonsson 2021), with respect to the context $C = \Pi$ of a planning program. The action novelty rank $r(a^*, \Pi)$ of an action $a^*$ is the count of the number of appearances of action $a^*$ in program $\Pi$. If action $a^* \notin \Pi$, then its rank is 1, whereas if action $a^*$ appears in every line of $\Pi$, then its rank is $n + 1$, where $n$ is the number of lines in $\Pi$.

## Structural Restrictions

We adopt two structural restrictions over the space of programs to keep the search space tractable without sacrificing completeness: 1) RAM actions `clear`, `dec` and `set` are not allowed in the first line, as they induce an unnecessary initial search plateau over the pointers, and 2) the destination line of a `goto` instruction (Segovia-Aguas, Jiménez, and Jonsson 2021) is not allowed to be another `goto`. One `goto` instruction can represent the same logic.

## Novelty-Based PGP

Novelty-based PGP, PGP($v$), uses the action novelty rank and a bound $v$ to prune a newly generated program $\Pi_{w_i=a^*}$ when $r(a^*, \Pi) > v$ where $\Pi_{w_i=a^*}$ is the planning program resulting from assigning action $a^*$ to the current programmable line $i$ in $\Pi$. Based on *empirical* reasoning, we fix the $v$ = 2 in PGP($v$) that we have submitted into the learning track of the IPC for both *agile* and *satisficing* metrics. Besides, PGP($v$) adopt the structural restrictions introduced above. Detailed explanation for PGP($v$) can be found in Lei, Lipovetzky, and Ehinger (2023)

## References


Lei, C.; Lipovetzky, N.; and Ehinger, K. A. 2023. Novelty and Lifted Helpful Actions in Generalized Planning. In *Proceedings of the 16th International Symposium on Combinatorial Search*, SoCS.

Lipovetzky, N.; and Geffner, H. 2012. Width and Serialization of Classical Planning Problems. In *Proceedings of the 20th European Conference on Artificial Intelligence*, ECAI, 540–545.

Segovia-Aguas, J.; Celorrio, S. J.; Sebastiá, L.; and Jonsson, A. 2022. Scaling-up generalized planning as heuristic search with landmarks. In *Proceedings of the 15th International Symposium on Combinatorial Search*, SoCS, 171–179.

Segovia-Aguas, J.; Jiménez, S.; and Jonsson, A. 2019. Computing programs for generalized planning using a classical planner. *Artificial Intelligence* 272: 52–85.

Segovia-Aguas, J.; Jiménez, S.; and Jonsson, A. 2021. Generalized planning as heuristic search. In *Proceedings of the 31st International Conference on Automated Planning and Scheduling*, ICAPS, 569–577.