Optimized Execution of PDDL Plans using Behavior Trees

AAMAS 2021 LONDON, UK

Francisco Martín
francisco.rico@urjc.es

Matteo Morelli
Matteo.MORELLI@cea.fr

Huascar Espinoza
Huascar.ESPINOZA@cea.fr

Francisco J.R. Lera
fjrodl@unileon.es

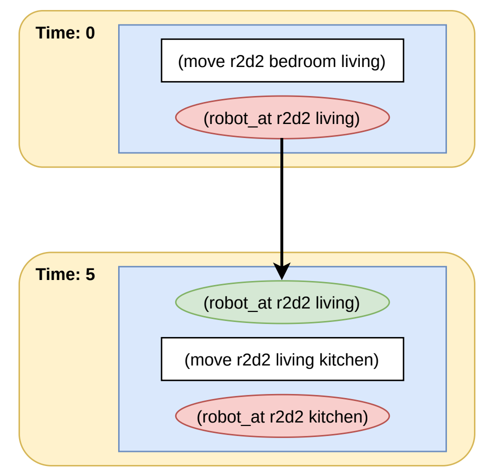
Vicente Matellán
vicente.matellan@unileon.es

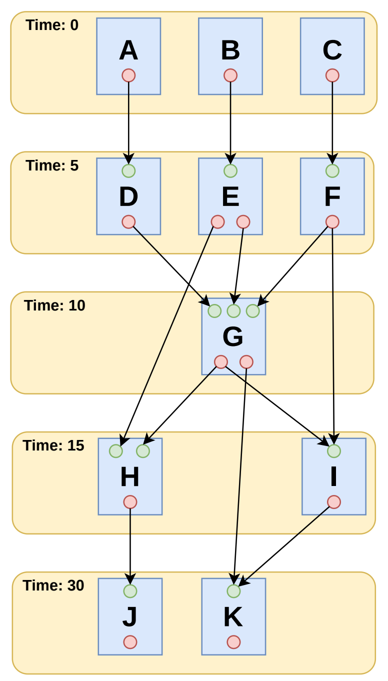
Intelligent Robotics Lab · Universidad Rey Juan Carlos · cea tech list · Grupo de Robótica · Universidad de León

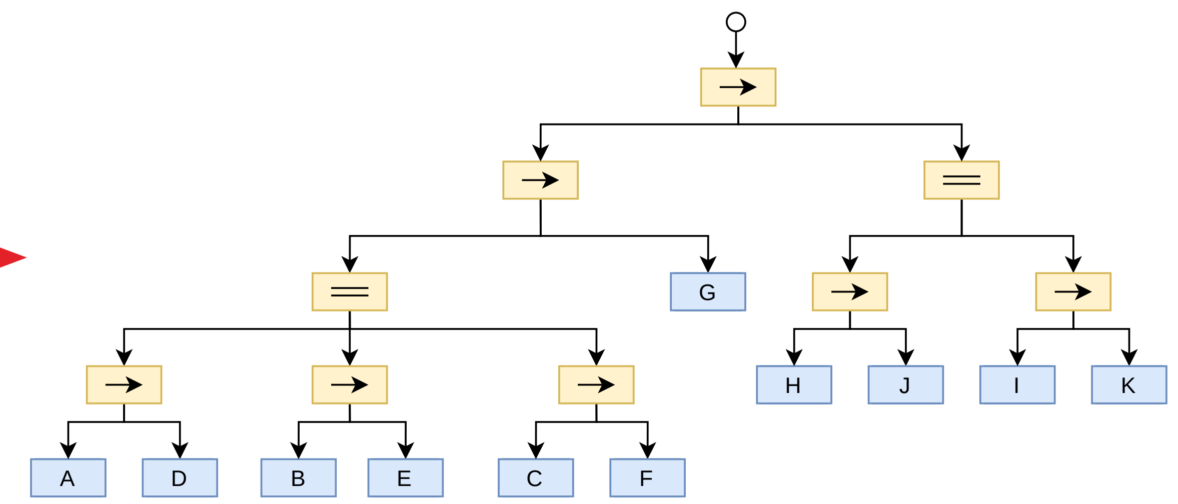
Robots need task planning to sequence and execute actions toward achieving their goals. On the other hand, Behavior Trees provide a mathematical model for specifying plan execution in an intrinsically composable, reactive, and robust way. PDDL (Planning Domain Definition Language) has become the standard description language for most planners. In this paper, we present a novel algorithm to systematically create behavior trees from PDDL plans to execute them. This approach uses the execution graph of the plan to generate a behavior tree. The most remarkable contribution of this approach is the algorithm to build a Behavior Tree that optimizes its execution by paralyzing actions, applicable to any plan, taking into account the actions' causal relationships. We demonstrate the improvement in the execution of plans in mobile robots using the ROS2 Planning System framework.

Goal: Convert PDDL Plans to Behavior Trees

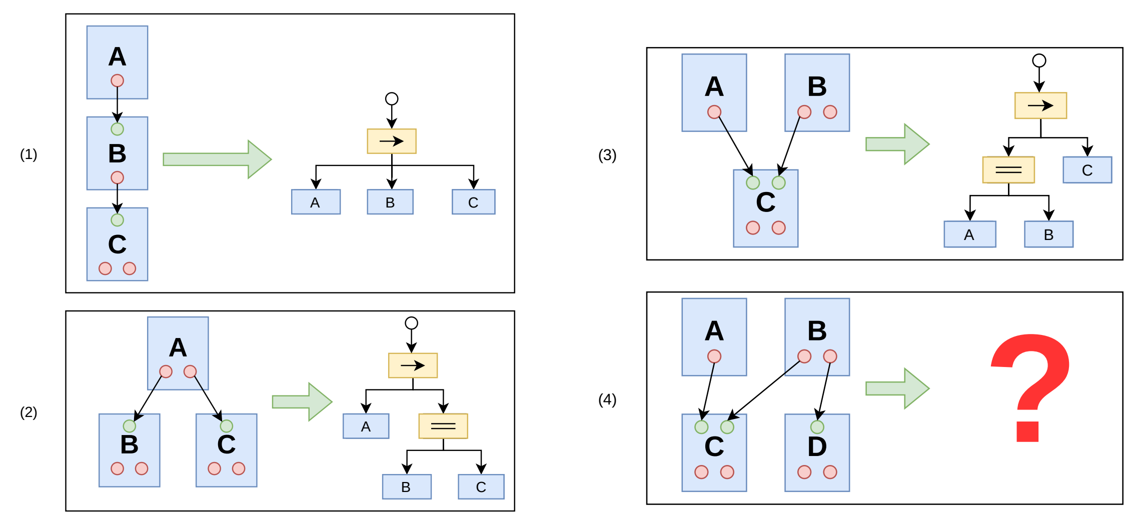
```
0        (move rb1 assembly_zone body_car_zone)
0        (move rb2 assembly_zone steerwheel_zone)
0        (move rb3 assembly_zone wheels_zone)
5.001    (transport rb1 bc_1 body_car_zone assembly_zone)
5.001    (transport rb2 stwhl_1 steerwheel_zone assembly_zone)
5.001    (transport rb3 whl_1 wheels_zone assembly_zone)
10.002   (assemble rb1 assembly_zone whl_1 bc_1 stwhl_1 car_1)
10.002   (move rb2 assembly_zone body_car_zone)
10.002   (move rb3 assembly_zone steerwheel_zone)
15.003   (move rb1 assembly_zone wheels_zone)
15.003   (transport rb2 bc_2 body_car_zone assembly_zone)
15.003   (transport rb3 stwhl_2 steerwheel_zone assembly_zone)
20.004   (transport rb1 whl_2 wheels_zone assembly_zone)
20.004   (move rb3 assembly_zone body_car_zone)
25.005   (assemble rb2 assembly_zone whl_2 bc_2 stwhl_2 car_2)
25.005   (move rb1 assembly_zone steerwheel_zone)
25.005   (transport rb3 bc_3 body_car_zone assembly_zone)
30.006   (move rb2 assembly_zone wheels_zone)
30.006   (transport rb1 stwhl_3 steerwheel_zone assembly_zone)
35.007   (transport rb2 whl_3 wheels_zone assembly_zone)
40.008   (assemble rb1 assembly_zone whl_3 bc_3 stwhl_3 car_3)
```



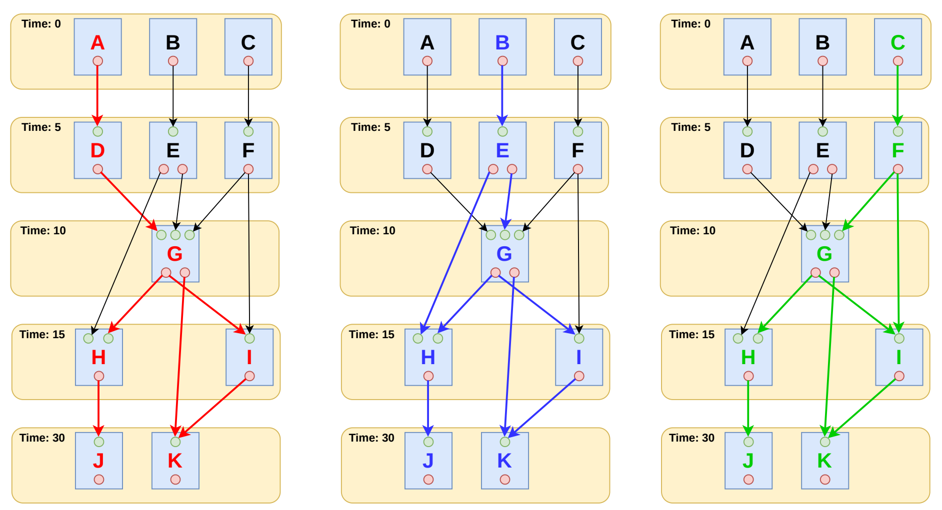




Problem: Not always possible



1 Execution Flows



Code

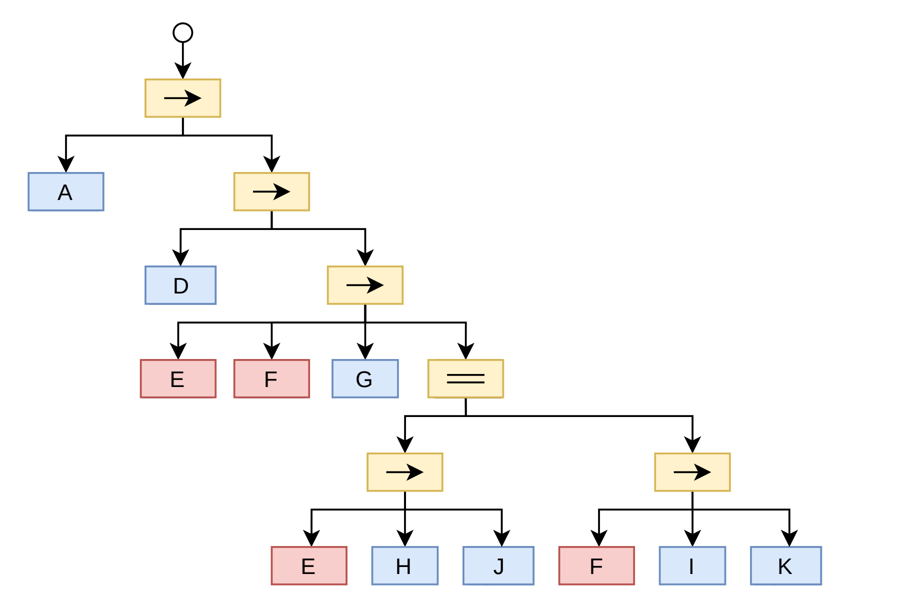
Full Paper

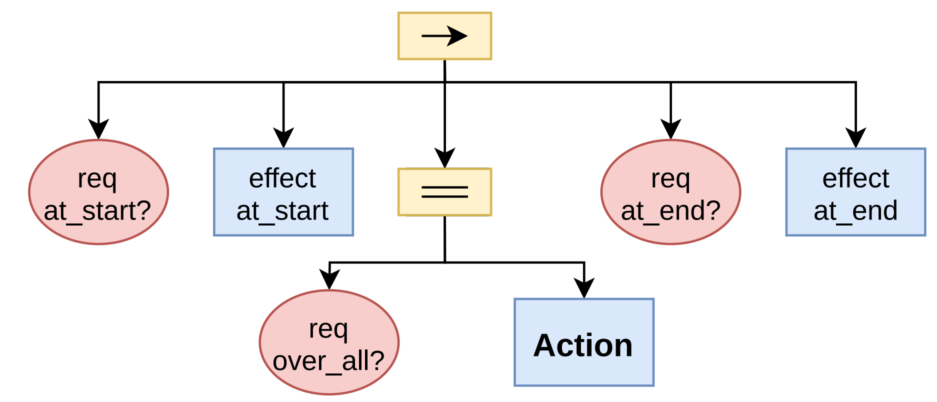
2 Algorithm

Flows ← $a_i \in A$, if $R_{a_i} = \emptyset$
if $|Flows| > 1$ **then**
  tree ← Parallel(get_tree($a_i, \emptyset$), $\forall a_i \in Flows$))
else
  tree ← get_tree($a_i, \emptyset$), $a_i \in Flows$)
end if
function GET_TREE(a, US)
  $US \leftarrow US \cup a$
  $W \leftarrow a_j \in A$, if $a_j \in R_a, a_j \notin US$
  $Succ \leftarrow a_j \in A$, if $a_j \neq a, a_j \in E_a$
  if $|a \rightarrow| == 0$ **then**
    if $W \neq \emptyset$ **then**
      return Sequence($Wait(a_j)$, Action(a)), $\forall a_j \in W$
    else
      return Action(a)
    end if
  else if $|a \rightarrow| == 1$ **then**
    return Sequence$\begin{pmatrix} Wait(a_j), \forall a_j \in W \\ Action(a), \\ get_tree(a_j, US), \forall a_j \in Succ \end{pmatrix}$
  else if $|a_i \rightarrow| > 1$ **then**
    return Sequence$\begin{pmatrix} Wait(a_j), \forall a_j \in W \\ Action(a), \\ Parallel(get_tree(a_j, US), \forall a_j \in Succ) \end{pmatrix}$
  end if

end function

3 Generated BT

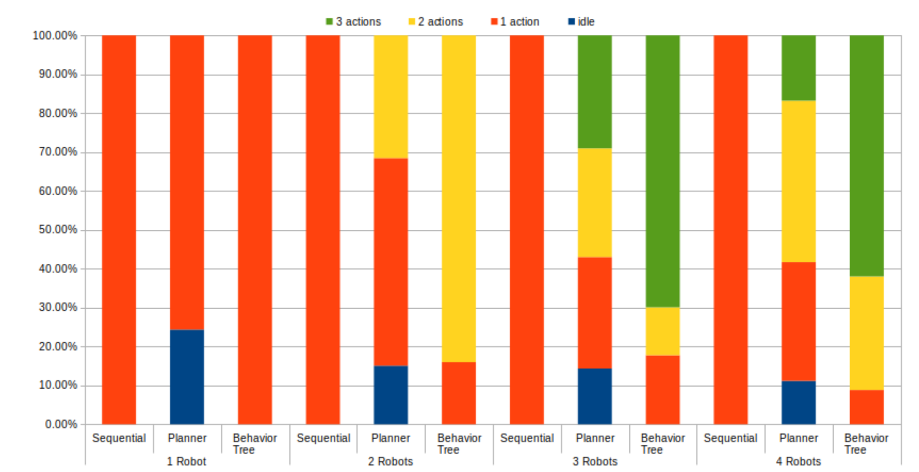


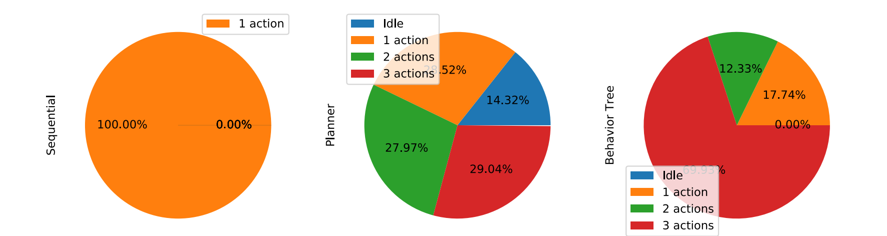


4 Validation Framework

PlanSys2

5 Experimental Results





This paper presents a proposal for using Behavior Trees to execute plans generated by a PDDL-based AI planner. Coding a plan as Behavior Tree is a compact way to represent and execute a robot action plan. Major contribution of the paper is the algorithm capable of transforming any plan into a Behavior Tree in a systematic way. This solution creates a planning graph from the plan and makes the tree recursively. Different types of nodes are used to build the Behavior Tree such as the singleton action node and the wait node to improve the efficiency of parallel excution of actions. The generated Behavior Tree is so optimized to execute in parallel all the possible actions in a plan, preserving the causal relationships of the actions. Another contribution is the execution an action as soon as its requirements are available, even before established in the plan.
This algorithm and the Behavior Trees executor have been included in the plan execution module of the ROS2 Planning System. Besides, it has the ability to execute plans on multiple robots collaboratively showing a positive impact in a real competition test with a multirobot variant.