**Results and Discussions**

This chapter presents the results of an analysis of train scheduling using temporal planning. The train scheduling domain was developed using PDDL 2.1 and PDDL 2.2. We used the LPG-TD automated temporal planner for simulation purposes. Because, as we previously stated, LPG-TD can handle both PDDL2.1 and PDDL2.2.

**Experiment:** We have experimented with our model in three different operating modes. Different operating modes give different results.

-n operating mode: -n operating mode gives us flexibility to choose the number of plans we want to generate. In our experiment, we considered generating 2 plans. And we get the following result:

*./lpg-td -o domain\_temporal.pddl -f problem\_temporal.pddl -n 2*

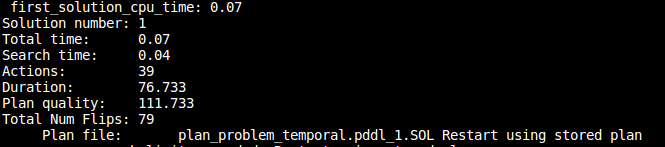
**f**

Fig: First solution

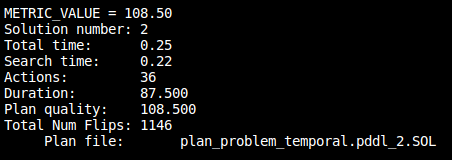
****

Fig: Second Solution

-Quality operating mode: This operating mode can give an optimal solution. It is mostly focused on quality and it ensures the best solution.

*./lpg-td -o domain\_temporal.pddl -f problem\_temporal.pddl -quality*

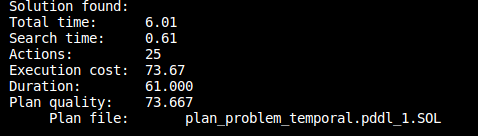
****

Fig: Quality mode solution

- This mode can provide a quick but not optimal solution. It can add an unnecessary step to give a quick solution.

*./lpg-td -o domain\_temporal.pddl -f problem\_temporal.pddl -speed*

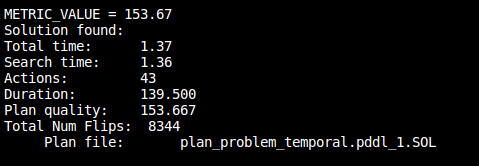
****

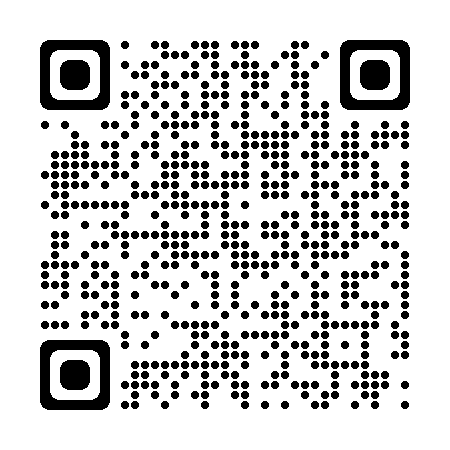
Fig: speed mode solution

The figures show the results of our experiment. So, after experimenting with the model we designed, we can say that for optimal train schedule, the -quality operating mode is the best choice.

**MACHINE CONFIGURATION**

|  |  |
| --- | --- |
| Operating System | 70.04 LTS |
| Processor | Intel(R) Core (TM) i5-7200U CPU @ 2.50GHz 2.70 GHz |
| RAM | 8.00 GB |
| SSD | 240 GB |
| System | 64-bit operating system, x64 – based processor |

All resources and documentation are available on github. The Git repository of our work is: <https://github.com/AminKaiser/Thesis-Temporal-Planning/tree/main/Version-2>

****