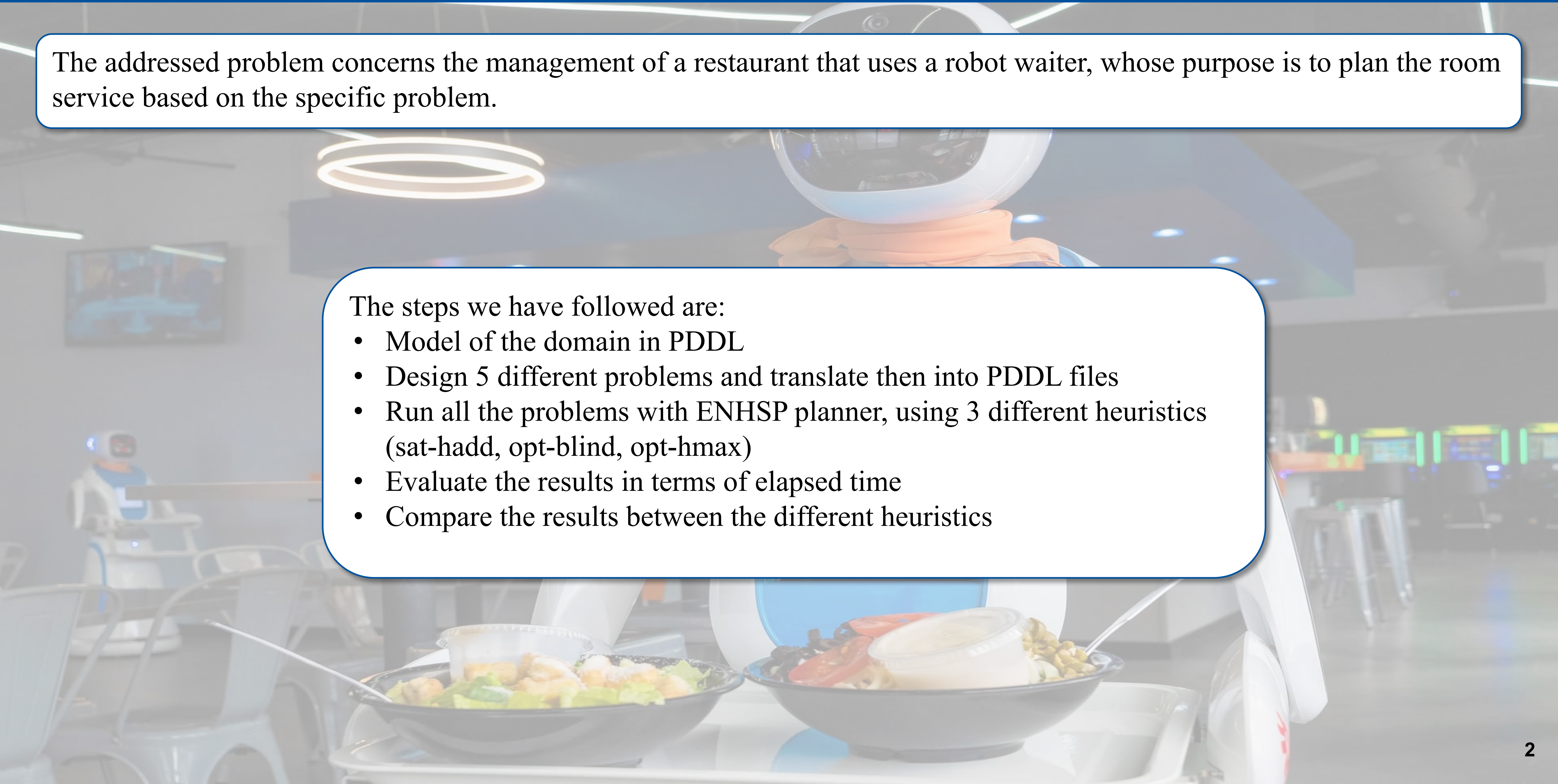
# PLANNING AND REASONING

**a.a 2023/2024**

# PROJECT PRESENTATION

Lorenzo Gizzi 1907374
Lorenzo Russo 2091186

# Introduction

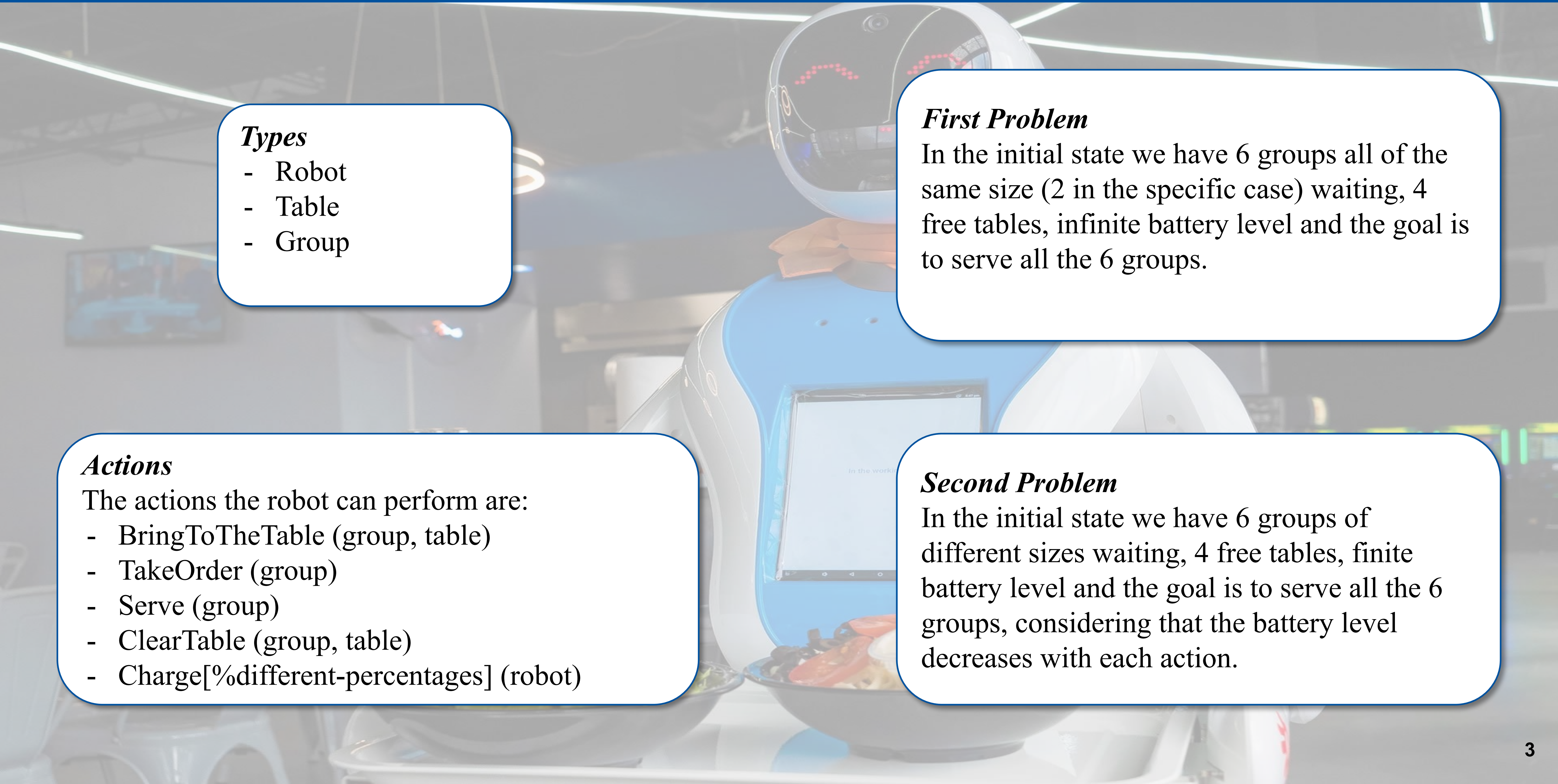


The addressed problem concerns the management of a restaurant that uses a robot waiter, whose purpose is to plan the room service based on the specific problem.

The steps we have followed are:

- Model of the domain in PDDL
- Design 5 different problems and translate then into PDDL files
- Run all the problems with ENHSP planner, using 3 different heuristics (sat-hadd, opt-blind, opt-hmax)
- Evaluate the results in terms of elapsed time
- Compare the results between the different heuristics

# Domain And Problems



***Types***

- Robot
- Table
- Group

***First Problem***

In the initial state we have 6 groups all of the same size (2 in the specific case) waiting, 4 free tables, infinite battery level and the goal is to serve all the 6 groups.

***Actions***

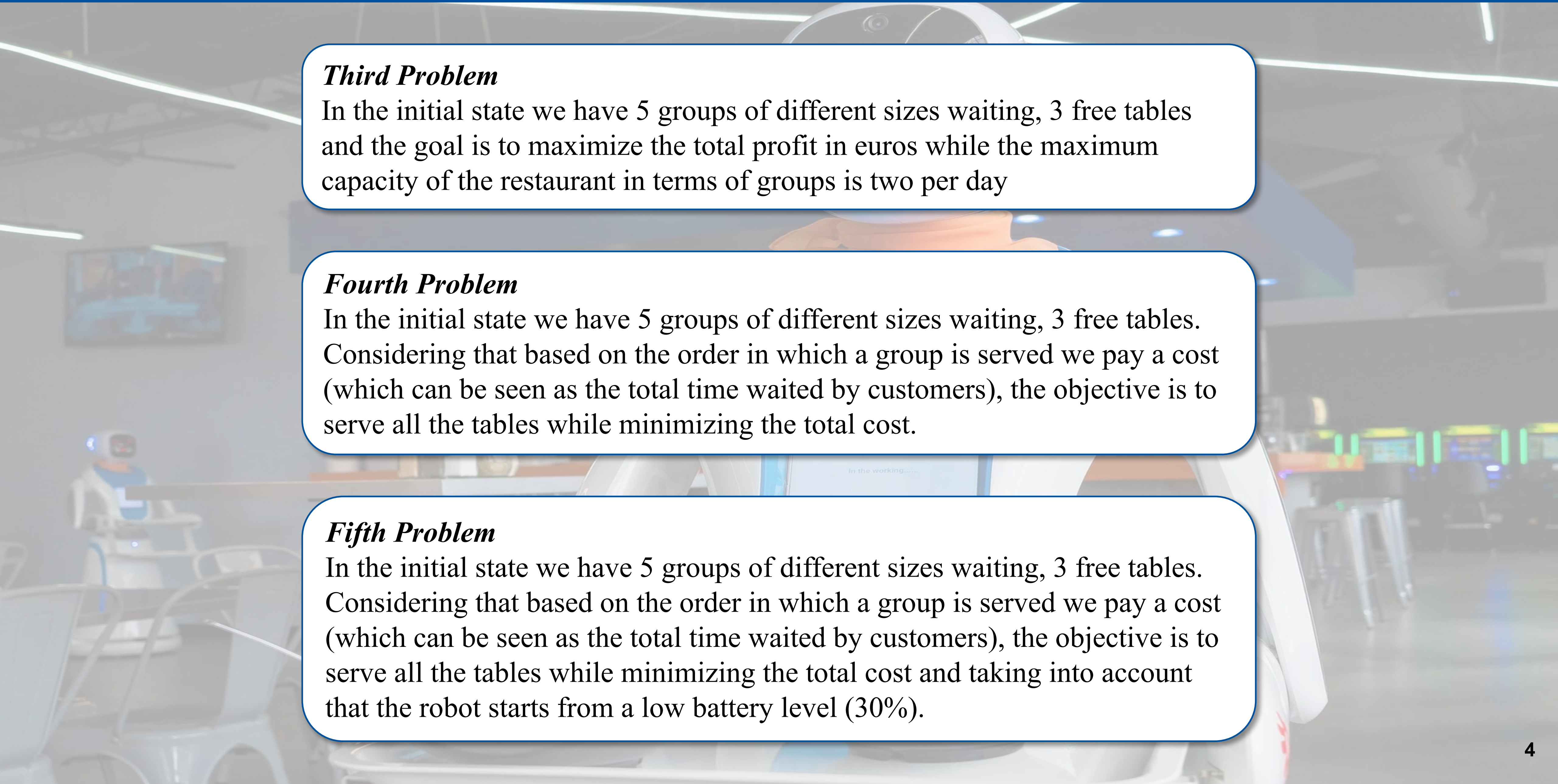
The actions the robot can perform are:

- BringToTheTable (group, table)
- TakeOrder (group)
- Serve (group)
- ClearTable (group, table)
- Charge[%different-percentages] (robot)

***Second Problem***

In the initial state we have 6 groups of different sizes waiting, 4 free tables, finite battery level and the goal is to serve all the 6 groups, considering that the battery level decreases with each action.

# Domain And Problems



***Third Problem***

In the initial state we have 5 groups of different sizes waiting, 3 free tables and the goal is to maximize the total profit in euros while the maximum capacity of the restaurant in terms of groups is two per day

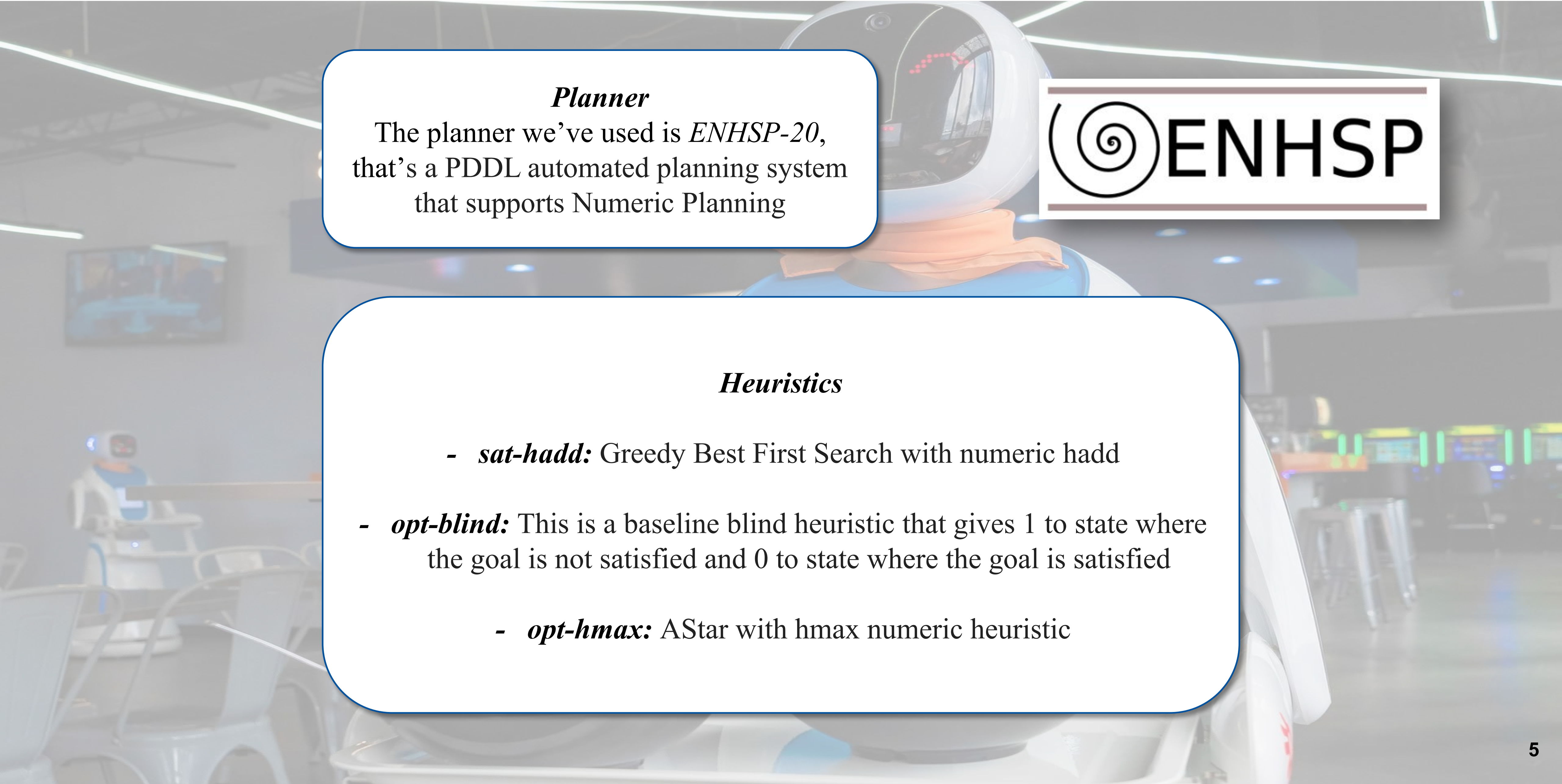
***Fourth Problem***

In the initial state we have 5 groups of different sizes waiting, 3 free tables. Considering that based on the order in which a group is served we pay a cost (which can be seen as the total time waited by customers), the objective is to serve all the tables while minimizing the total cost.

***Fifth Problem***

In the initial state we have 5 groups of different sizes waiting, 3 free tables. Considering that based on the order in which a group is served we pay a cost (which can be seen as the total time waited by customers), the objective is to serve all the tables while minimizing the total cost and taking into account that the robot starts from a low battery level (30%).

# Experiments



***Planner***

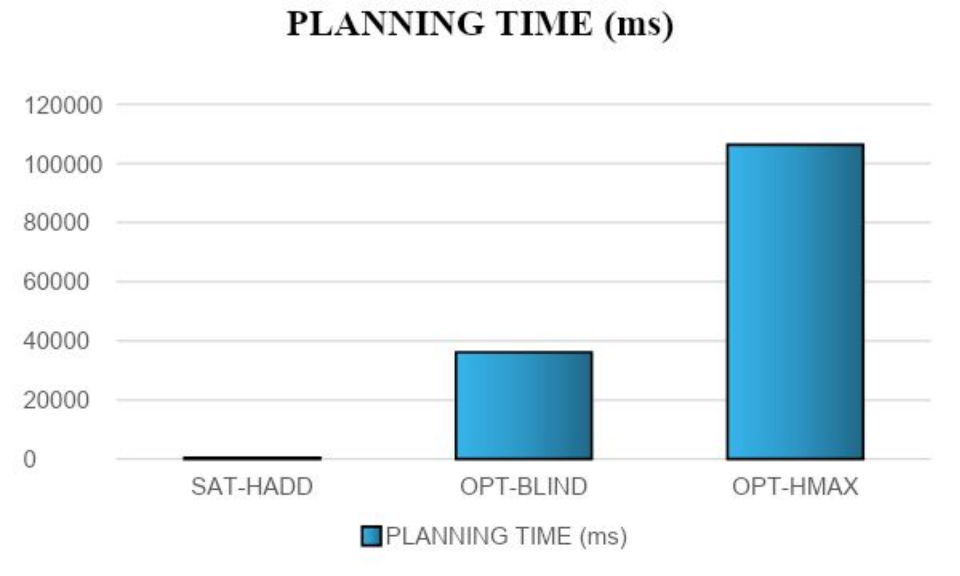
The planner we've used is *ENHSP-20*, that's a PDDL automated planning system that supports Numeric Planning

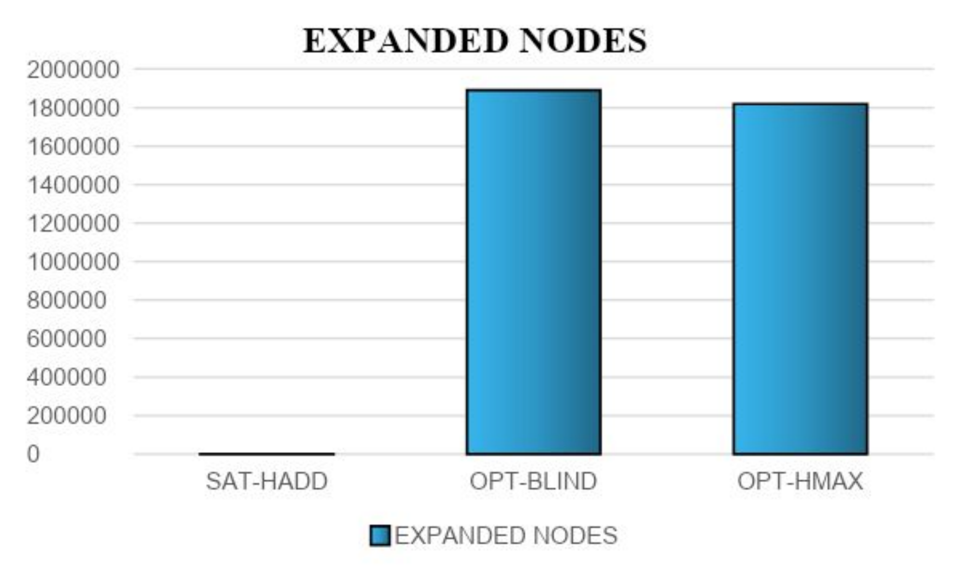
ENHSP

***Heuristics***

- ***sat-hadd:*** Greedy Best First Search with numeric hadd
- ***opt-blind:*** This is a baseline blind heuristic that gives 1 to state where the goal is not satisfied and 0 to state where the goal is satisfied
- ***opt-hmax:*** AStar with hmax numeric heuristic

# Analysis – First Problem

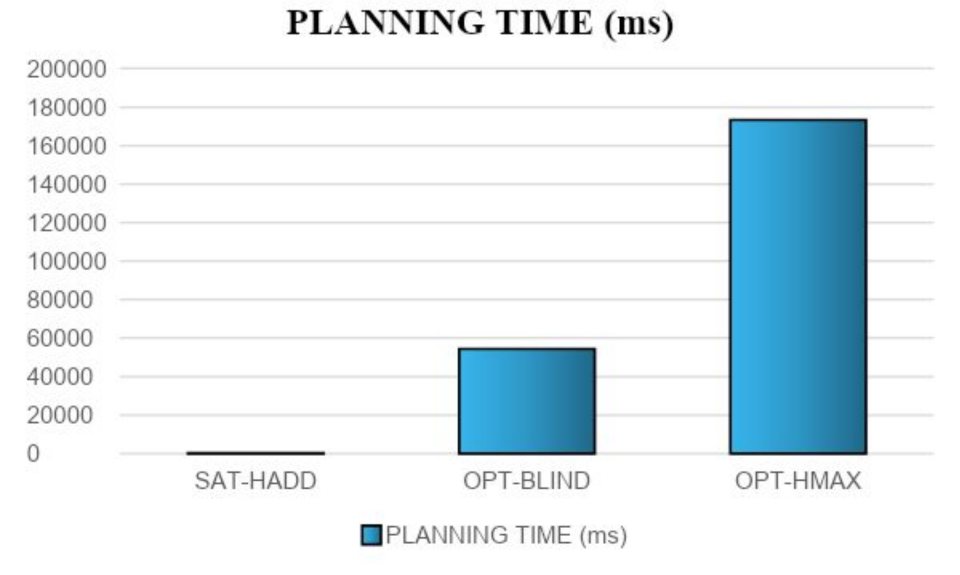


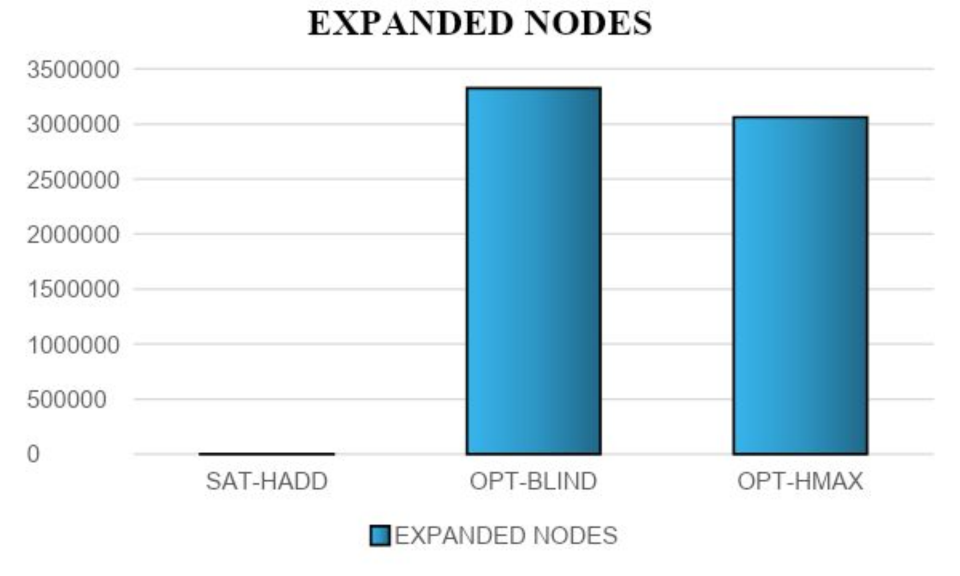


***Initial state:***
6 groups
4 tables
infinite battery
***Goal:***
serve all the groups

***Opt-hmax plan:***
*(bringToTheTable1 g6 t2)*
*(bringToTheTable2 g1 t3)*
*(bringToTheTable3 g5 t1)*
*(bringToTheTable4 g4 t4)*
*(takeOrder g4)*
*(serve g4)*
*(takeOrder g1)*
*(serve g1)*
*(clearTable g4 t4)*
*(bringToTheTable5 g3 t4)*
*(takeOrder g3)*
*(serve g3)*
*(clearTable g1 t3)*
*(bringToTheTable5 g2 t3)*
*(takeOrder g2)*
*(clearTable g3 t4)*
*(serve g2)*
*(takeOrder g5)*
*(clearTable g2 t3)*
*(serve g5)*
*(takeOrder g6)*
*(serve g6)*
*(clearTable g6 t2)*
*(clearTable g5 t1)*

# Analysis – Second Problem

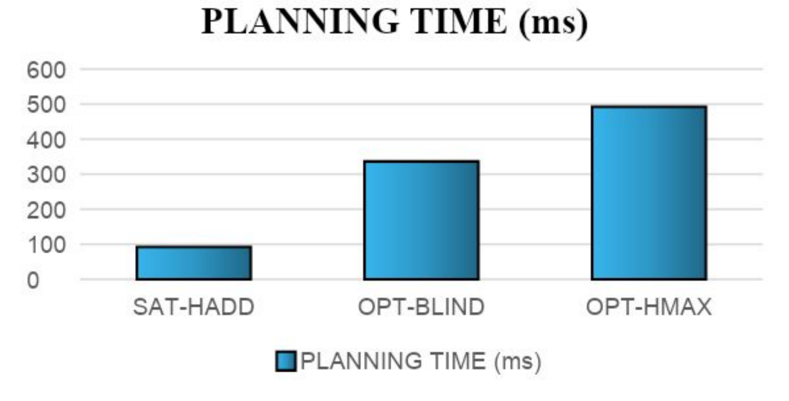


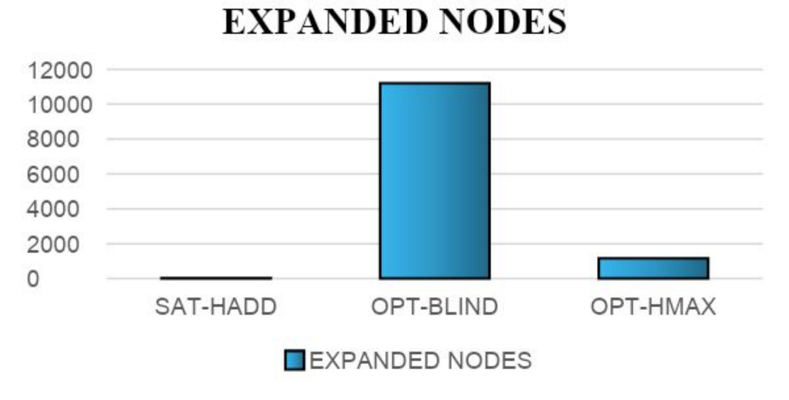


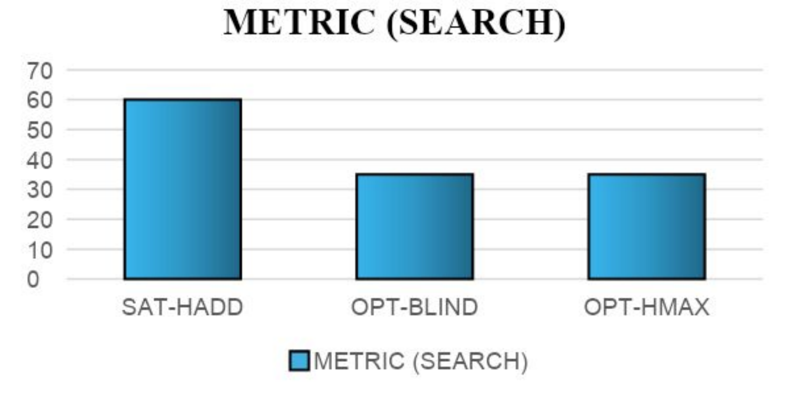
***Initial state:***
6 groups
4 tables
battery level 100%
***Goal:***
serve all the groups

***Opt-hmax plan:***
*(bringToTheTable1 g1 t3)*
*(bringToTheTable2 g4 t4)*
*(bringToTheTable3 g6 t2)*
*(takeOrder g1)*
*(serve g1)*
*(takeOrder g6)*
*(clearTable g1 t3)*
*(serve g6)*
*(bringToTheTable4 g3 t3)*
*(takeOrder g3)*
*(clearTable g6 t2)*
*(bringToTheTable5 g5 t2)*
*(takeOrder g5)*
*(takeOrder g4)*
*(charge100 self)*
*(serve g4)*
*(serve g3)*
*(clearTable g4 t4)*
*(clearTable g3 t3)*
*(serve g5)*
*(bringToTheTable5 g2 t3)*
*(takeOrder g2)*
*(serve g2)*
*(clearTable g2 t3)*
*(clearTable g5 t2)*

# Analysis – Third Problem


PLANNING TIME (ms)

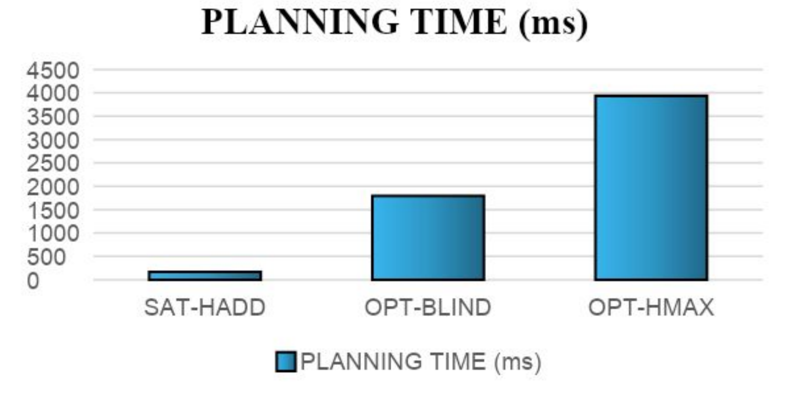

EXPANDED NODES

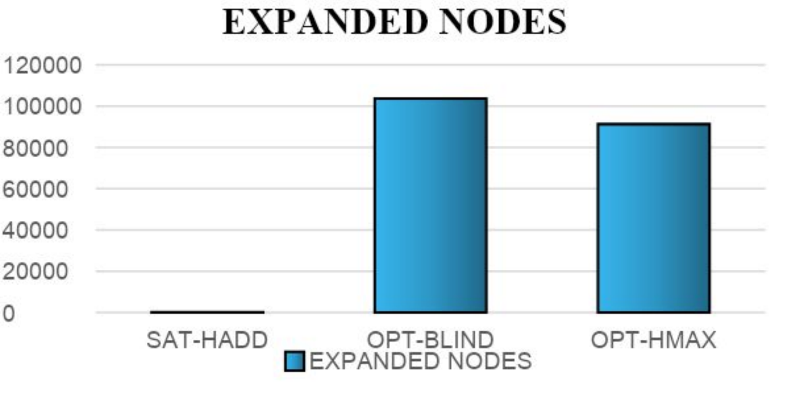

METRIC (SEARCH)

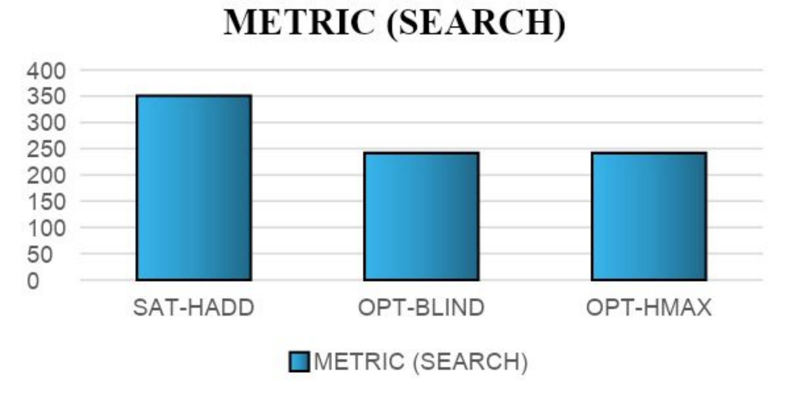
***Initial state:***
5 groups
3 tables
battery level 100%
capacity of max 2 groups daily
***Goal:***
serve 2 groups
***Metric:***
maximize total profit

***Opt-hmax plan:***
*(bringToTheTable1 g3 t3)*
*(bringToTheTable2 g4 t2)*
*(takeOrder g4)*
*(serve g4)*
*(takeOrder g3)*
*(serve g3)*
*(clearTable g4 t2)*
*(clearTable g3 t3)*

# Analysis – Fourth Problem



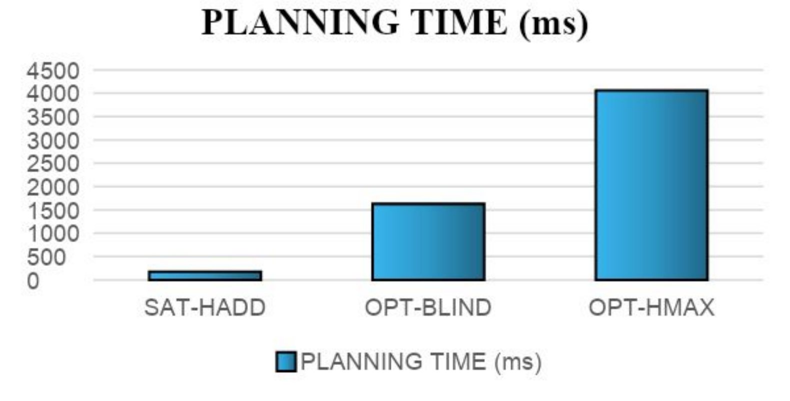


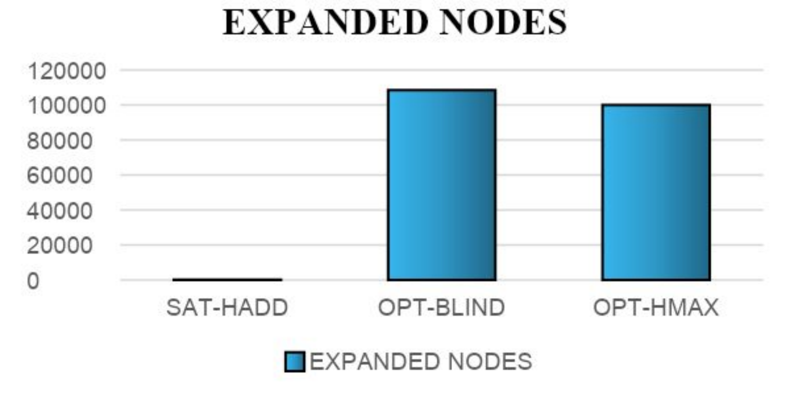


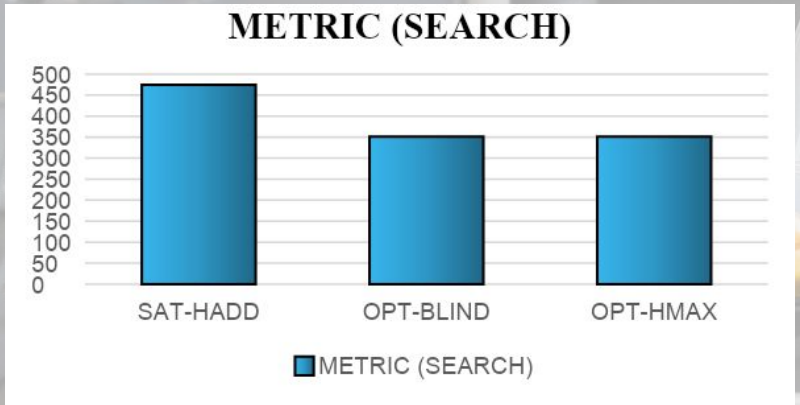
***Initial state:***
5 groups
3 tables
battery level 100%
***Goal:***
serve all the groups
***Metric:***
minimize total cost

***Opt-hmax plan:***
*(bringToTheTable1 g1 t1)*
*(bringToTheTable2 g3 t3)*
*(bringToTheTable3 g4 t2)*
*(takeOrder g1)*
*(serve g1)*
*(takeOrder g4)*
*(clearTable g1 t1)*
*(bringToTheTable4 g2 t1)*
*(takeOrder g2)*
*(serve g2)*
*(takeOrder g3)*
*(charge60 self)*
*(serve g4)*
*(clearTable g4 t2)*
*(serve g3)*
*(clearTable g3 t3)*
*(bringToTheTable5 g5 t2)*
*(takeOrder g5)*
*(serve g5)*
*(clearTable g5 t2)*
*(clearTable g2 t1)*

# Analysis – Fifth Problem



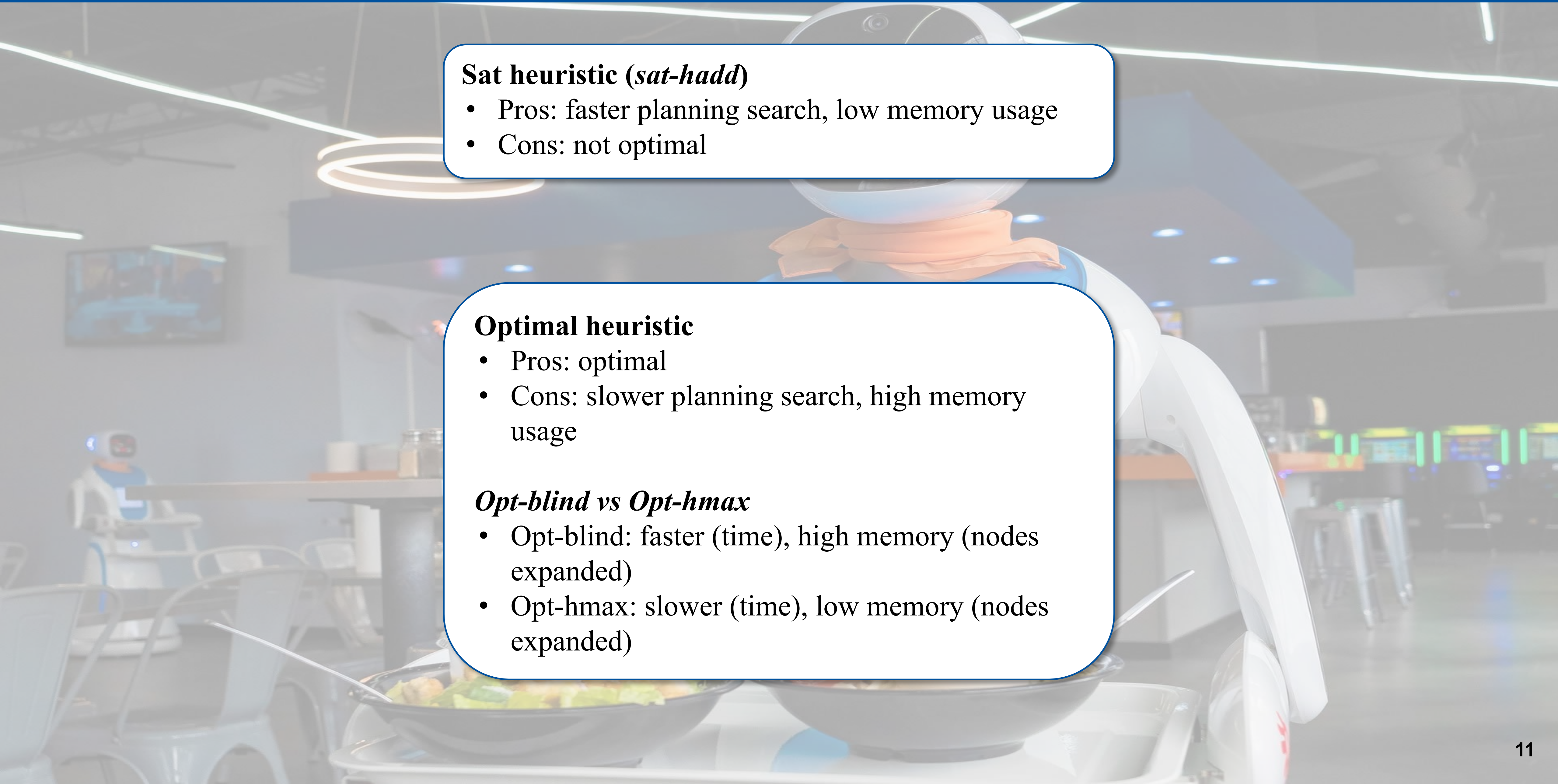




***Initial state:***
5 groups
3 tables
battery level 30%
***Goal:***
serve all the groups
***Metric:***
minimize total cost

***Opt-hmax plan:***
*(bringToTheTable1 g2 t1)*
*(bringToTheTable2 g3 t3)*
*(bringToTheTable3 g4 t2)*
*(takeOrder g4)*
*(charge100 self)*
*(takeOrder g2)*
*(serve g2)*
*(serve g4)*
*(charge40 self)*
*(clearTable g2 t1)*
*(bringToTheTable4 g1 t1)*
*(takeOrder g1)*
*(takeOrder g3)*
*(clearTable g4 t2)*
*(serve g1)*
*(serve g3)*
*(clearTable g3 t3)*
*(bringToTheTable5 g5 t2)*
*(takeOrder g5)*
*(serve g5)*
*(clearTable g5 t2)*
*(clearTable g1 t1)*

# Conclusion



**Sat heuristic (*sat-hadd*)**

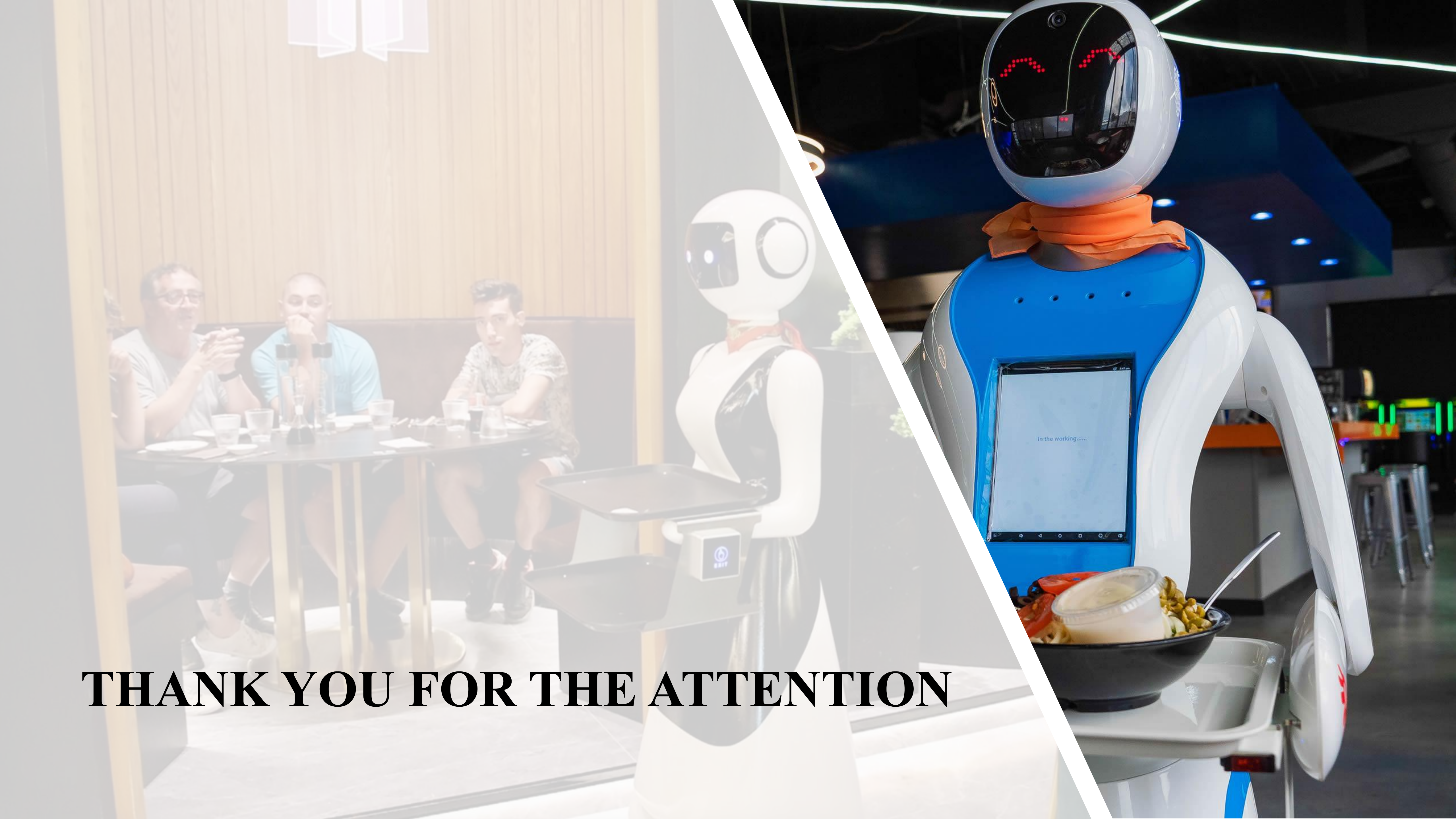
- Pros: faster planning search, low memory usage
- Cons: not optimal

**Optimal heuristic**

- Pros: optimal
- Cons: slower planning search, high memory usage

***Opt-blind vs Opt-hmax***

- Opt-blind: faster (time), high memory (nodes expanded)
- Opt-hmax: slower (time), low memory (nodes expanded)



# THANK YOU FOR THE ATTENTION