



# Rush Hour

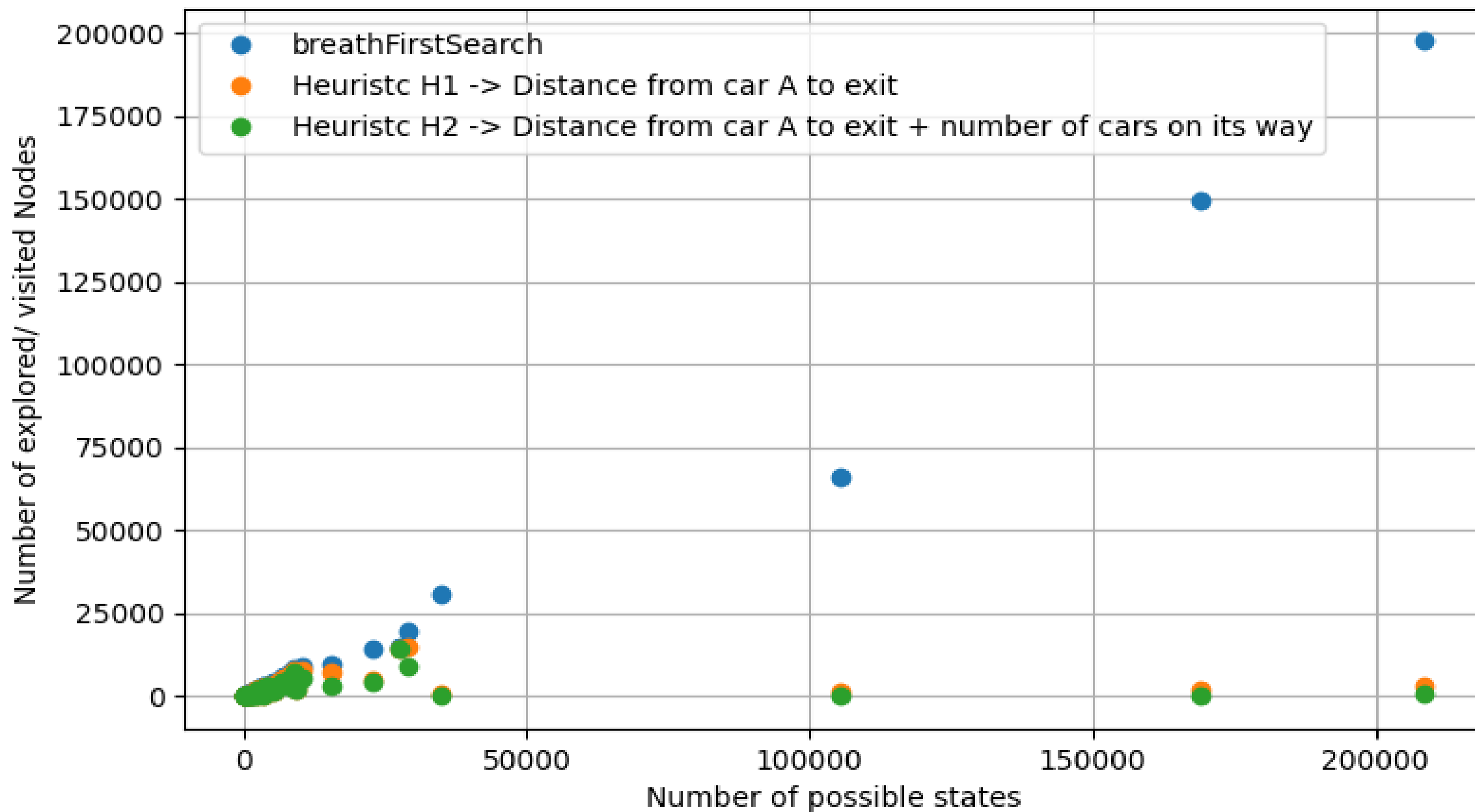
102435 RAFAEL REMIGIO

101043 MARCEL SOUZA

# Algoritmo Utilizado

- ▶ O Algoritmo utilizado foi o BestFirstSearch, pois utilizando as heurísticas como forma de resolução, visitamos muito menos nós, se comparado ao algoritmo anterior utilizado: Breath Search.
- ▶ A seguir, deixamos um gráfico comparando o número de nós visitados pelas diferentes heurísticas testadas durante o processo. ]
- ▶ Foram testadas 2 heurísticas diferentes (H1 e H2). Ao final, por melhor performance, optamos por utilizar a heurística H2.

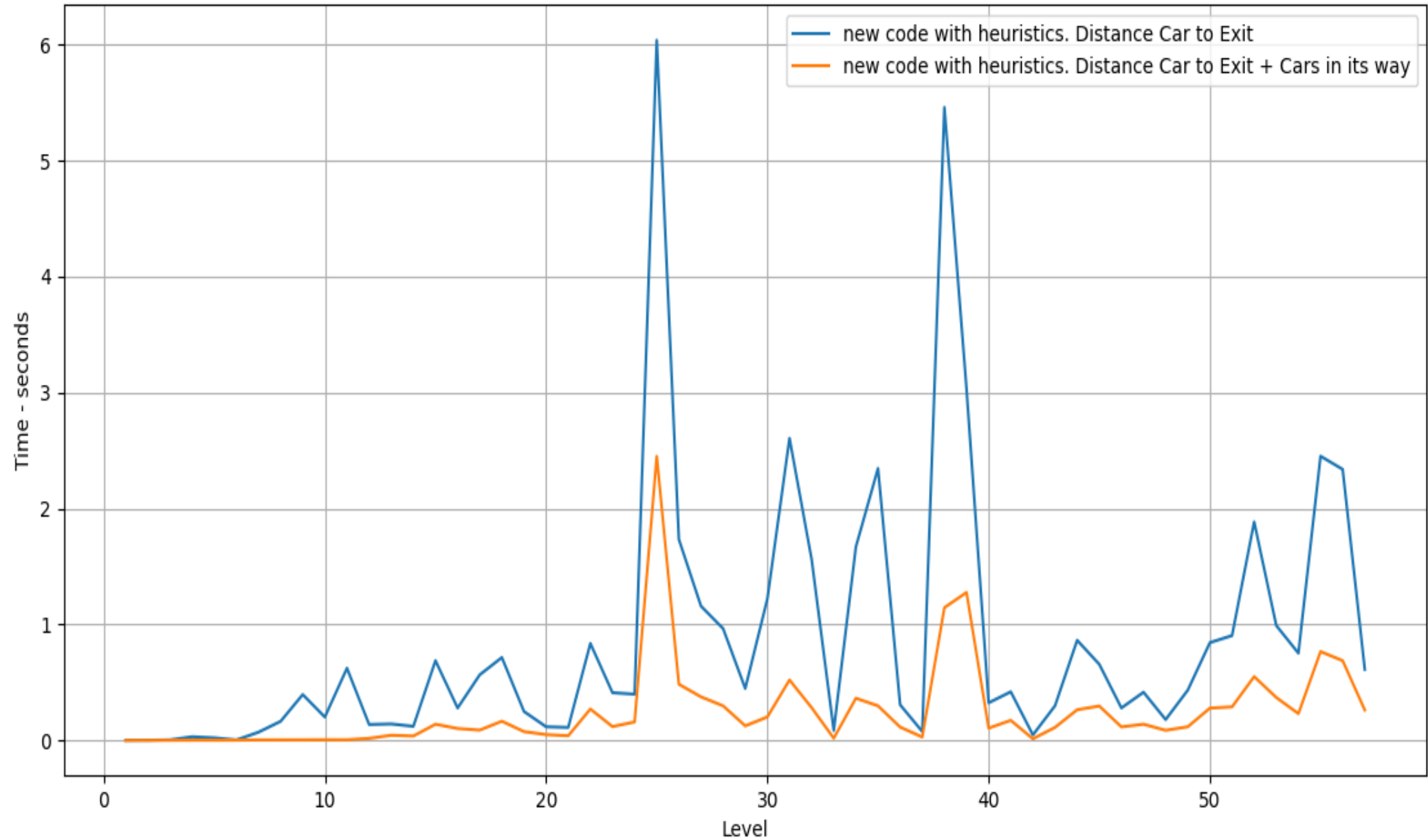
Comparing Number of visited Nodes!



# Heurísticas Utilizadas

- ▶ Foram testadas 2 heurísticas:
- ▶  $H1 = \text{Distância entre o carro e a saída};$
- ▶  $H2 = H1 + \text{número de carros no seu caminho até a saída};$
- ▶ O H2 se performa melhor do que o H1, afinal, se um carro tiver um caminho livre pela frente até a saída, visitará menos nós e, portanto, será mais rápido. Seguem alguns gráficos que comprovam o benefício da escolha pelo H2 em detrimento da heurística H1.

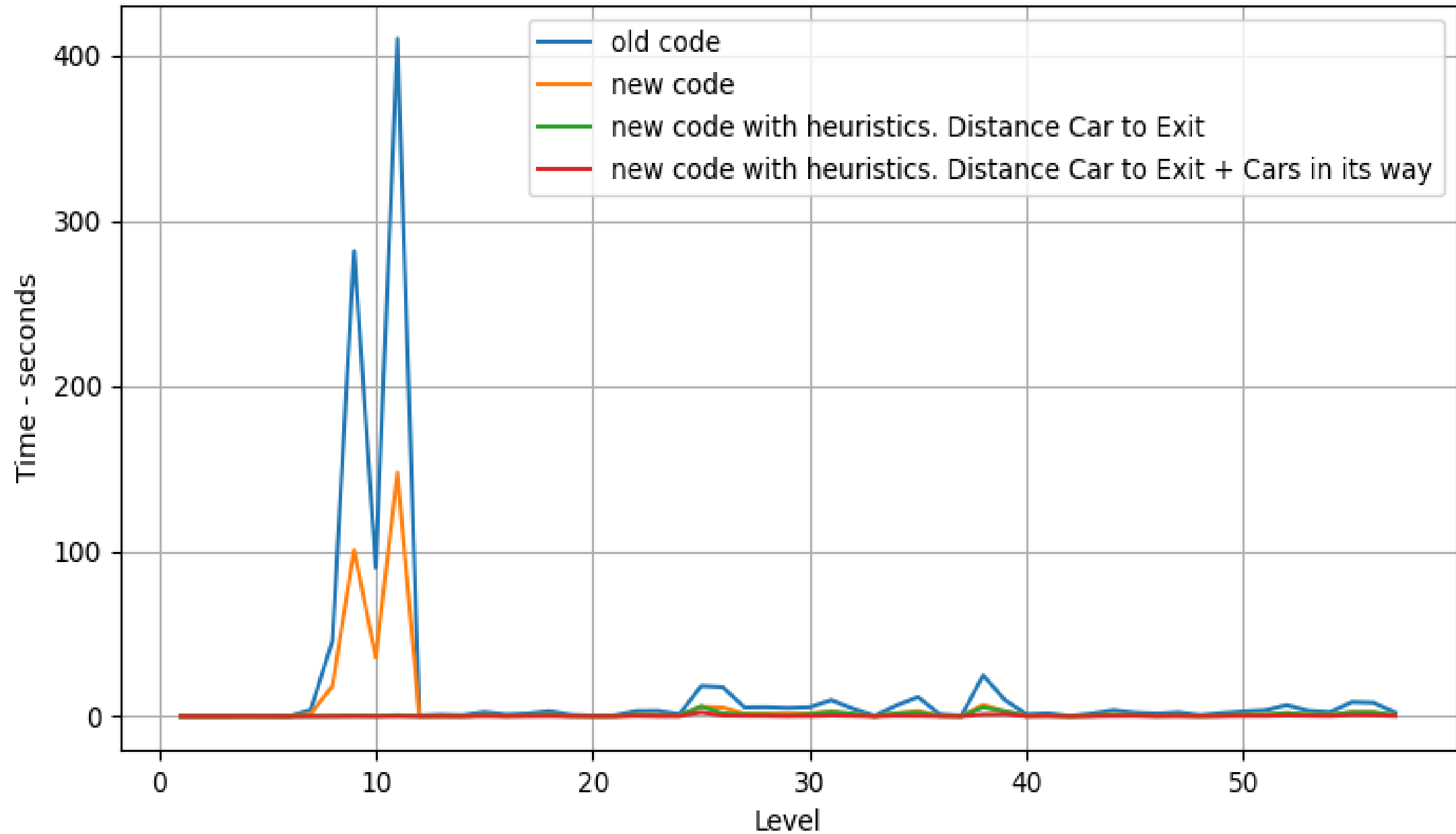
Code Changes Comparing times!



# Optimização

- ▶ Podemos ver no próximo gráfico, os efeitos de algumas optimizações feitas ao código da primeira entrega.
- ▶ OldCode= 1º entrega (Breath Search)
- ▶ NewCode= Melhorias (utilização de programação funcional e estruturas de dados alternativas (tuplos).
- ▶ NewCode with heuristics (H1 e H2) = Best First Search Algorithm juntamente com as heurísticas citadas anteriormente.
- ▶ Percebemos que a complexidade temporal diminui drasticamente (mesmo utilizando o Breath Search). Esta diminui ainda mais após utilizarmos as heurísticas e o Best First Search;

Code Changes Comparing times!



# Benchmarks

