UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Universidade do Algarve

Inteligência Artificial

Licenciatura em Engenharia Informática, 3º Ano, 1º Semestre, 2022/23

Problema 3: n-*queens*

IA2223P1G2:
Luís Ramos, 71179
Roberto Santos, 71214
Jorge Silva, 72480

Docente: José Valente de Oliveira

UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Índice

UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Descrição do problema

O problema *n-queens* é um problema clássico que consiste na colocação de n rainhas num tabuleiro de xadrez num formato n*n, sem que nenhuma rainha se ataque mutuamente.

N-queens é tipicamente um problema representativo de uma categoria de problemas economicamente relevantes.

No tabuleiro existe sempre uma rainha por coluna e uma por linha no máximo, tal como nas diagonais, no entanto existem diagonais sem nenhuma rainha.

UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

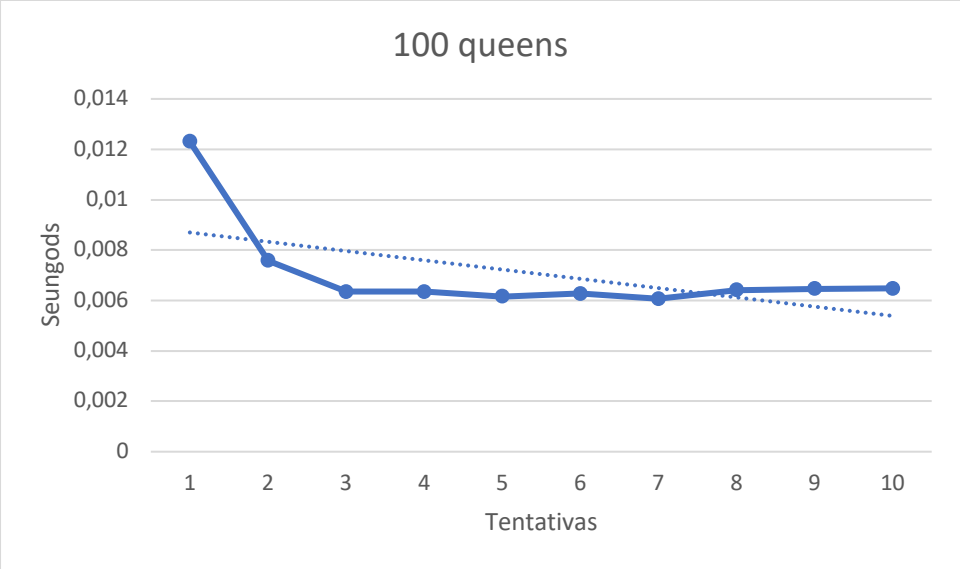
Metodologia

No problema proposto é nos dado a possibilidade de utilizar qualquer tipo de algoritmo de inteligência artificial lecionado nas aulas teóricas para a resolução do problema *n-queens*.
Após uma pesquisa intensiva sobre o problema, o algoritmo que se enquadrou melhor para as nossas necessidades foi uma variante do algoritmo *Local Search.*
Inicialmente durante as nossas pesquisas encontramos um estudo que explicava um algoritmo de inserção de rainhas no tabuleiro, de forma a que independentemente do tamanho do tabuleiro conseguia arranjar uma ou mais soluções (através de um *shift right* em todas as rainhas).
Mas devido a este algoritmo vir a ser demasiado específico e sua difícil implementação no *design Strategy Pattern* tivemos que descartar este algoritmo.
A razão da escolha de uma variante do algoritmo *Local Search* deve-se à possibilidade *do Local Search* padrão ficar preso no máximo/mínimo local. A forma de como resolvemos este problema, foi iterando várias vezes sobre o tabuleiro até não serem realizados mais movimentos de rainhas no tabuleiro, ou seja, todas as rainhas estarão na melhor posição possível.

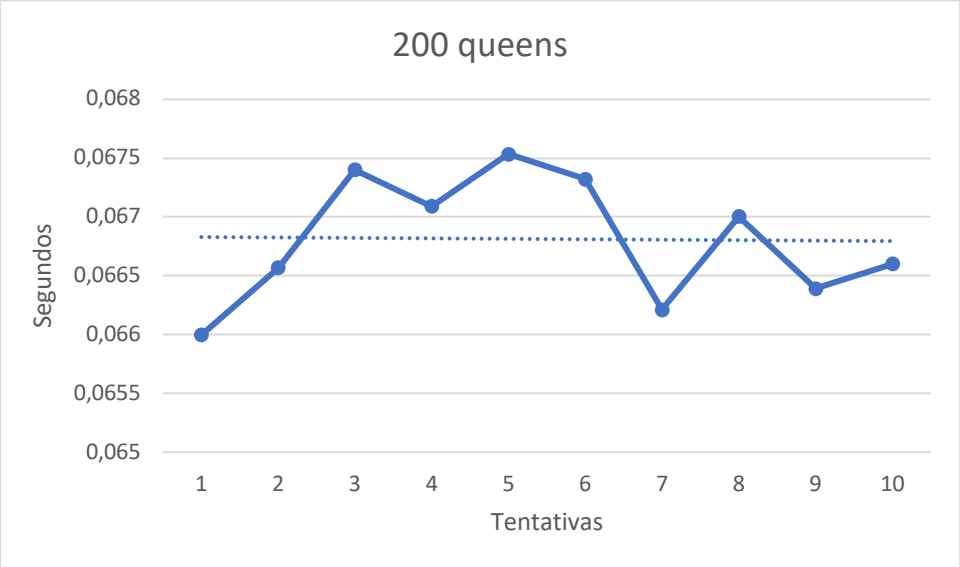
UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Casos de Teste

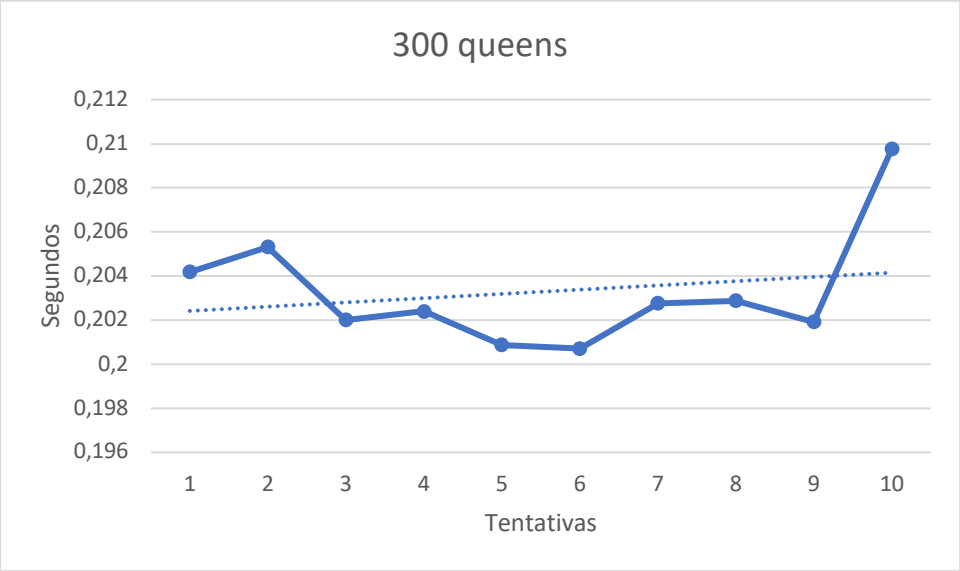
Foram realizados casos de teste de forma incremental, ou seja, começamos com um número relativamente baixo de rainhas e vamos aumentado sucessivamente o número de rainhas.
O objetivo destes testes é testar o tempo de execução e se o tabuleiro encontrado é uma solução.
Estes casos de testes poderão se encontrar dentro da pasta do projeto.



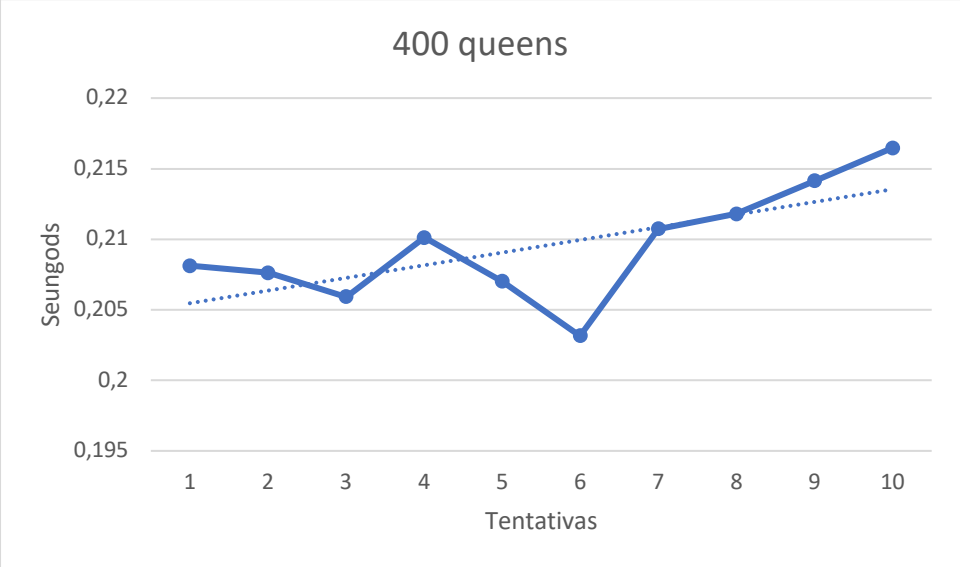
100 *queens*, resultados com base em média de 10 em variações de 100.



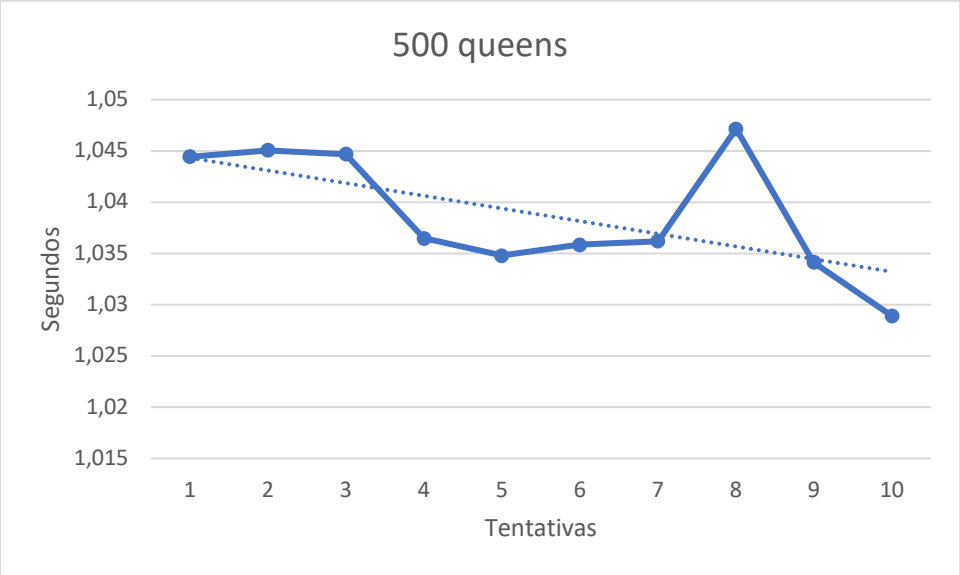
200 *queens*, resultados com base em média de 10 em variações de 100.



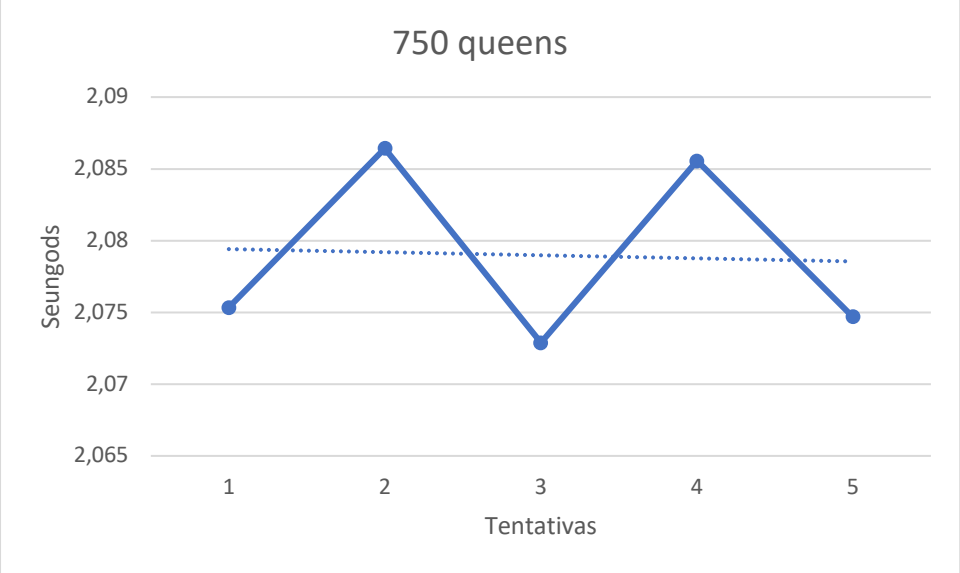
300 *queens*, resultados com base em média de 10 em variações de 100.



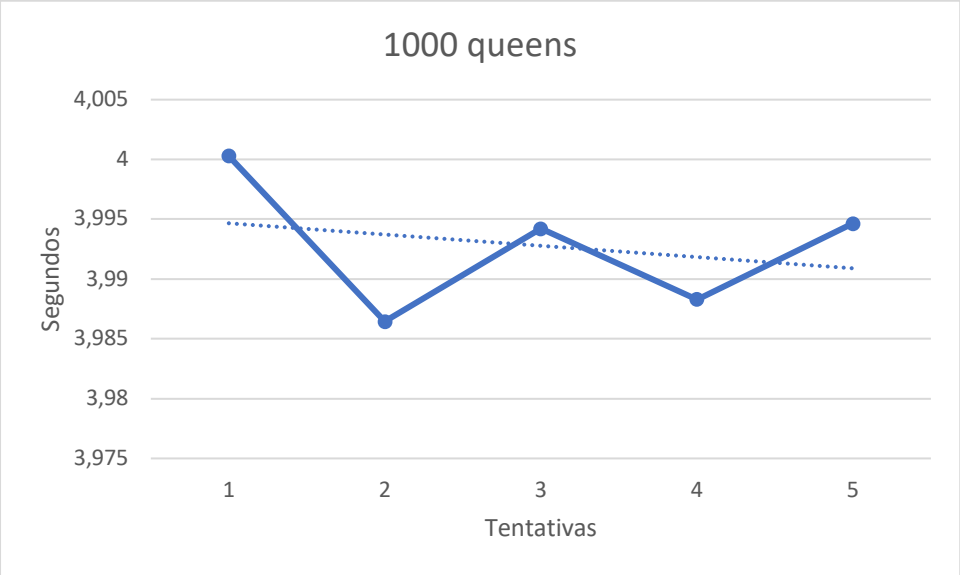
400 *queens*, resultados com base em média de 10 em variações de 100.



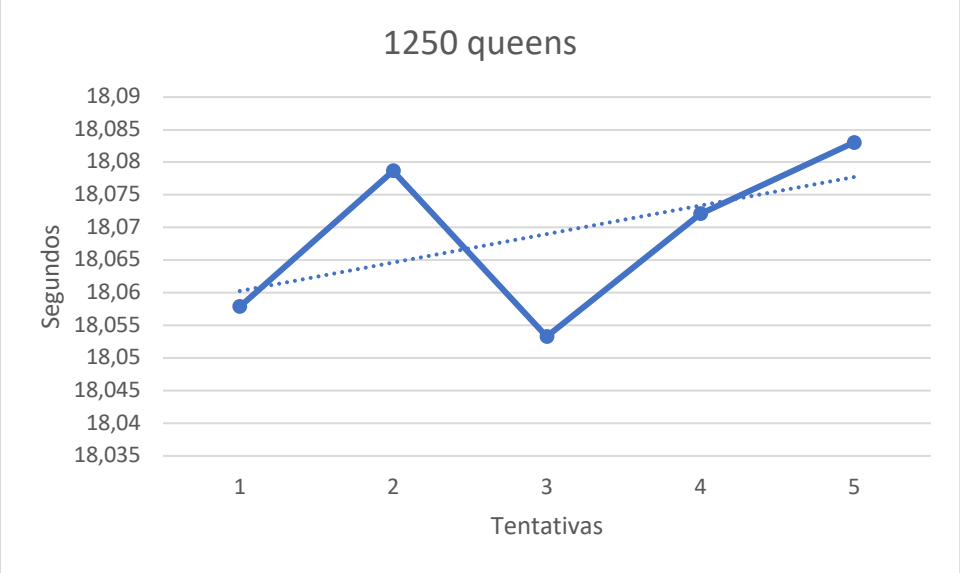
500 *queens*, resultados com base em média de 10 em variações de 100.



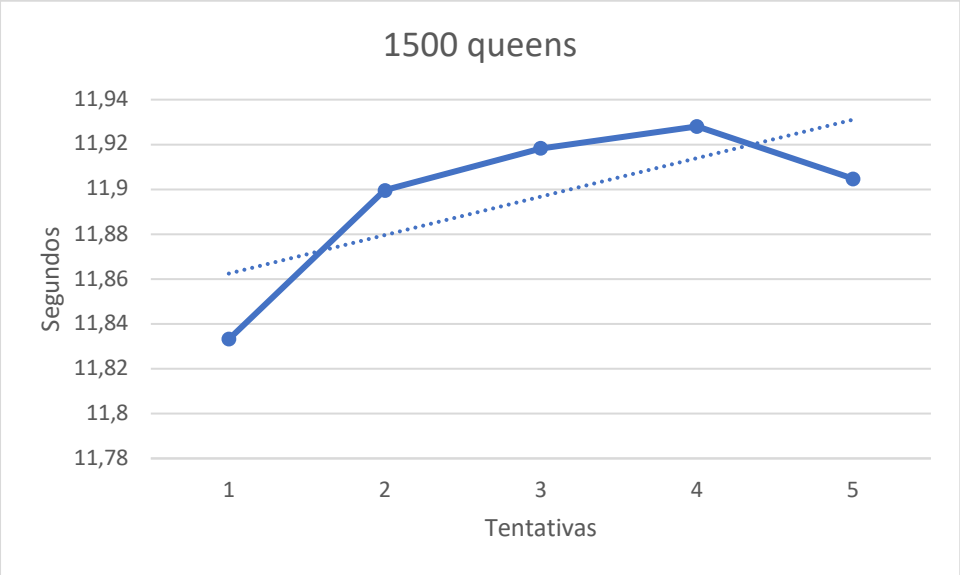
750 *queens*, resultados com base em média de 5 em variações de 250.



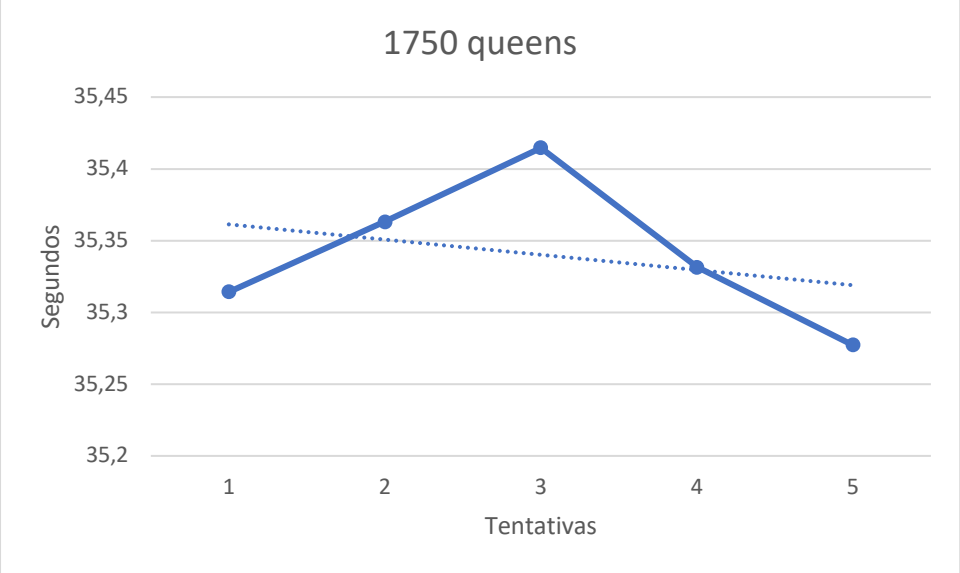
1000 *queens*, resultados com base em média de 5 em variações de 250.



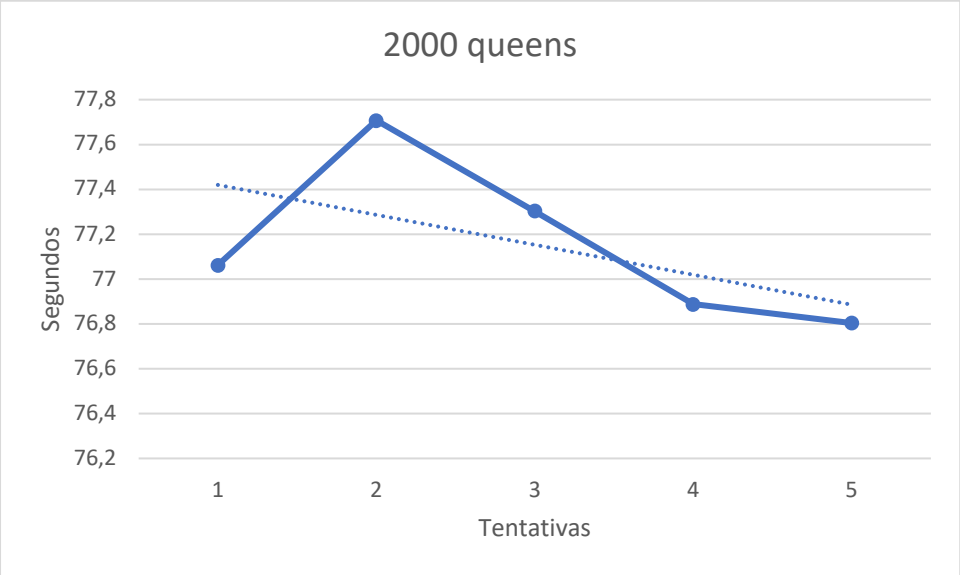
1250 *queens*, resultados com base em média de 5 em variações de 250.



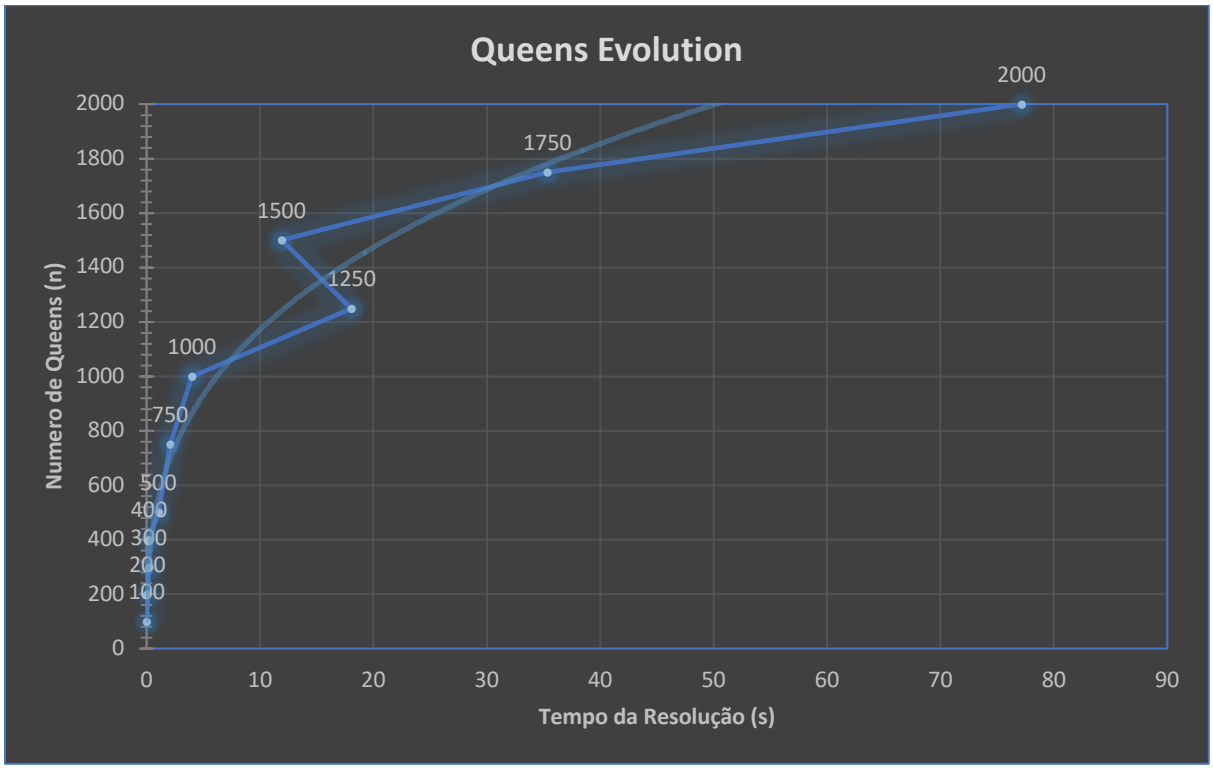
1500 *queens*, resultados com base em média de 5 em variações de 250.



1750 *queens*, resultados com base em média de 5 em variações de 250.



2000 *queens*, resultados com base em média de 5 em variações de 250.



O Gráfico (*Queens Evolution*) tem como objetivo facilitar a análise face a evolução da dificuldade do problema, neste caso o aumento do número de n (*queens*), através da média de tempo em s(segundos) obtida pelo nosso programa na resolução de cada fase.

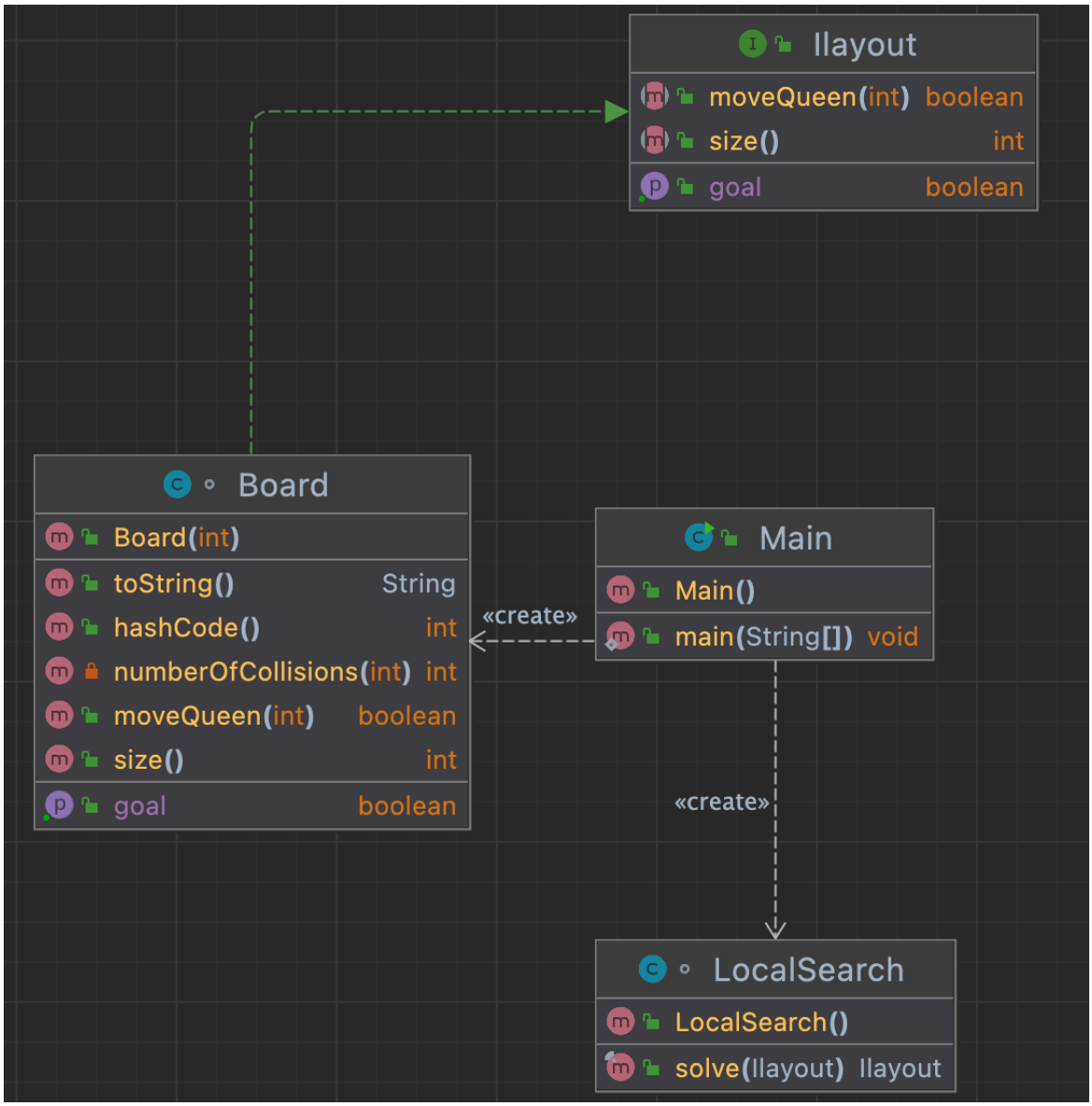
UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Padrões de projeto

No início o nosso maior problema foi quando ficávamos presos no máximo/mínimo local.
Após deparar-nos com este problema refletimos por eventuais soluções do problema, mais tarde solucionando-o com inspiração no algoritmo mencionado nas referências, que consiste em verificar todos os pares de rainhas e trocá-las se caso essa troca seja favorável, ou seja, se o número de colisões diminuir.
Ao ler este artigo surgiu a ideia de iterar várias vezes sobre o tabuleiro, com a condição que mesmo que tenha o mesmo número de colisões a rainha irá mover-se abrindo portas a outras rainhas que estavam "presas".
Agora falando sobre a implementação usamos uma permutação como representação do tabuleiro e dois *arrays* que representam as diagonais positiva e negativa. Para calcular as colisões, como temos um *array* com todas as diagonais, tanto positivas como negativas, só precisamos de verificar se existe alguma rainha na mesma linha e somar esse valor com o número de rainhas que existem nas diagonais. O cálculo das colisões é usado para escolhermos o melhor sítio para colocar uma rainha, escolhemos o sítio que tem melhor ou igual número de colisões e passamos para a próxima rainha, assim sucessivamente. Quando chegarmos ao fim da iteração, se não tivermos feito nenhuma troca, significa que as rainhas estão no melhor sítio possível, ou seja, é altura de testarmos se é uma solução

UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Diagrama de classes UML de implementação



UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Conclusão

Em suma, este problema permitiu-nos aprofundar conhecimentos sobre algoritmos de procura, de forma a otimizá-los o máximo possível.

Foi um problema bastante desafiante, que ficou melhor do que o inicialmente previsto. Não obstante nos termos deparado com algumas dificuldades para a finalização deste problema, nomeadamente, constrangimentos com a eficiência de números elevados, devido à complexidade do problema.

Podemos concluir, que apesar de não haver um objetivo padrão para a realização do problema, sentimos que alcançamos o nosso objetivo pessoal.

UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Referências

http://cse.unl.edu/~choueiry/Documents/Sosic-Gu-N-queens-1990.pdf