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 4: *Order-based GA*

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 do caixeiro viajante (também conhecido como problema do caminho mais curto) é um problema clássico da teoria dos grafos. Consiste em encontrar a rota mais curta possível para que um viajante visite uma série de cidades e retorne ao ponto de partida, passando por cada cidade apenas uma vez.

O problema pode ser formalizado da seguinte maneira: dado um conjunto de cidades e as distâncias entre elas, encontrar a rota mais curta que visite todas as cidades e volte ao ponto de partida.

UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Metodologia

No problema proposto é nos dada a possibilidade de utilizar qualquer tipo de algoritmo genético lecionado nas aulas teóricas para a resolução de qualquer problema que possa ser solucionado através desta implementação, usando o design *Strategy Pattern*.

Após uma pesquisa rápida sobre o problema, tornou-se evidente a necessidade da criação de uma interface para representar os indivíduos, para podermos utilizar o mesmo algoritmo e estratégia, mesmo que usando indivíduos com caraterísticas distintas. Para concretizar isto, todos os indivíduos têm que ter uma função de fitness, que é posteriormente para fazer a seleção dos indivíduos mais promissores, ou seja, que representem uma (esperançosamente boa) solução para o problema do caixeiro viajante.

UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Padrões de projeto

Inicialmente tentámos adaptar o projeto de *n-queens* com algoritmos genéticos, mas após diversas adversidades, decidimos mudar de problema e escolhemos o problema do Caixeiro viajante.

Neste problema, houve a necessidade de criar uma nova classe de suporte chamada *City*, que representa cada cidade. Esta classe contém as coordenadas de uma cidade e um método de cálculo de distância entre cidades. Para fazer este cálculo utilizamos a distância euclidiana.

Na *class Path*, que implementa a interface *IIndividual*, o cromossoma é um *array* de City's, ou seja, todas as cidades sobre as quais se pretende achar o caminho mais curto de forma a passar por todas as cidades apenas uma vez.

O cálculo do fitness de cada individuo é feito através da soma das distâncias entre cada par de cidades, pela ordem que aparecem no cromossoma. Como estamos à procura de um indivíduo com o melhor fitness possível, fizemos 1 sobre a distância total, o que nos dá a proporcionalidade desejada (quanto menor a distância, maior o valor da relação 1/totalDistance).

A mutação deste tipo de indivíduo é feita através de uma técnica chamada SBM (*Select Best Mutation*). Para concretizar este método, utilizámos 4 tipos de mutação (shift, inversion, swap, shuffle) e no fim escolhemos o que tem melhor fitness (mais potencial). Para fazer o crossoover, utilizámos o *CX (Cycle Crossover)*.

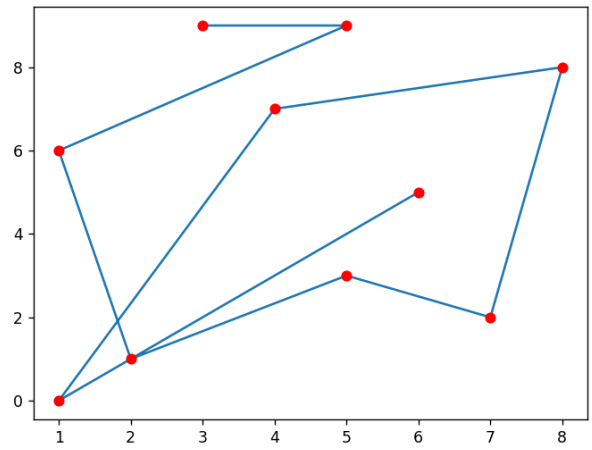
UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Casos de Teste /Análises dos resultados

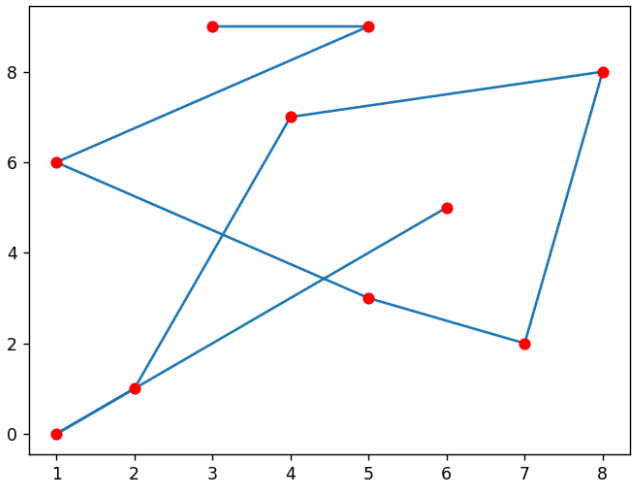
Foram realizados casos de teste com base na evolução das gerações, onde analisamos os melhores resultados de geração em geração. O objetivo destes testes é facilitar-nos a análise da evolução das gerações.

Trabalhando com uma população de 100 indivíduos e 10 cidades, vamos escolher as gerações onde o melhor resultado foi aprimorado, ou seja, houve um aumento significativo no fitness. Utilizando a probabilidade 0.25 de mutação e 0.25 de crossover:

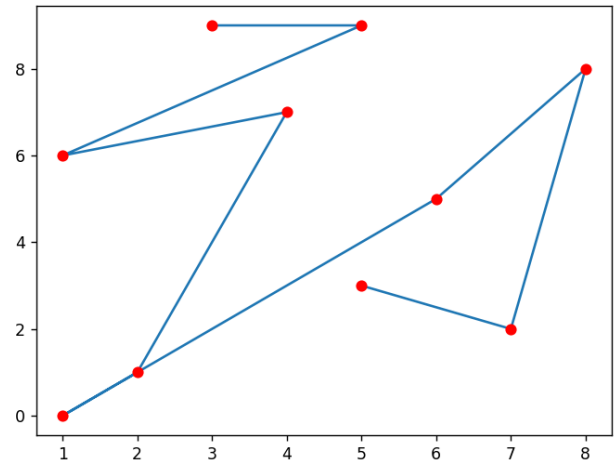
Geração 1: fitness 0.0233462



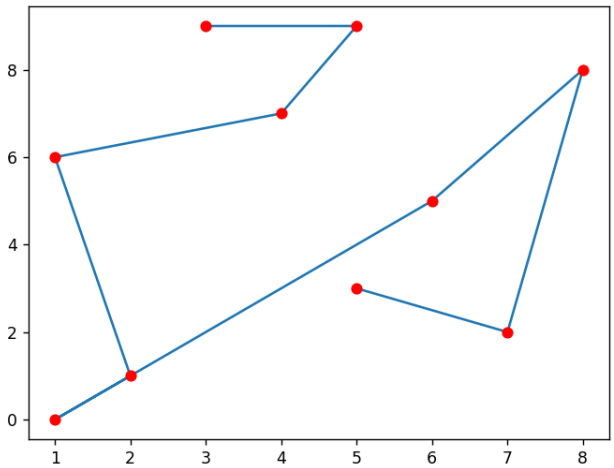
Geração 6: fitness 0.0255566



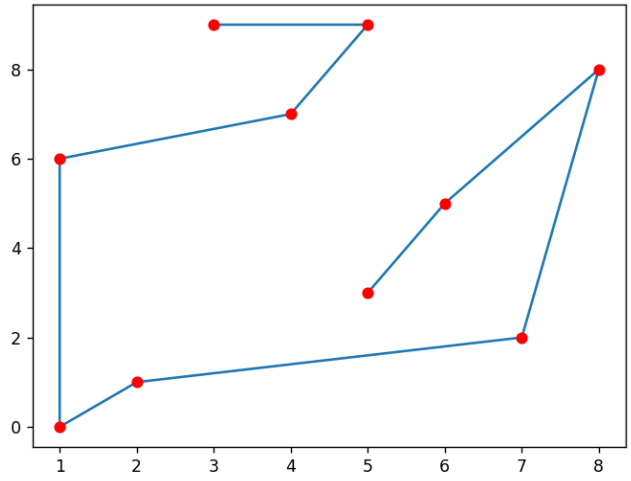
Geração 7: fitness 0.02710284



Geração 11: fitness 0.3038864



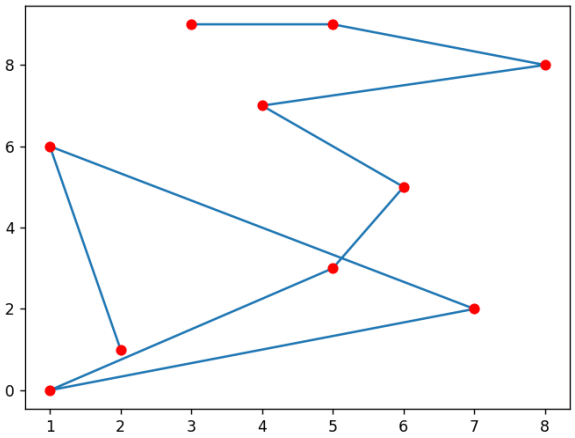
Geração 13: fitness 0.03141102



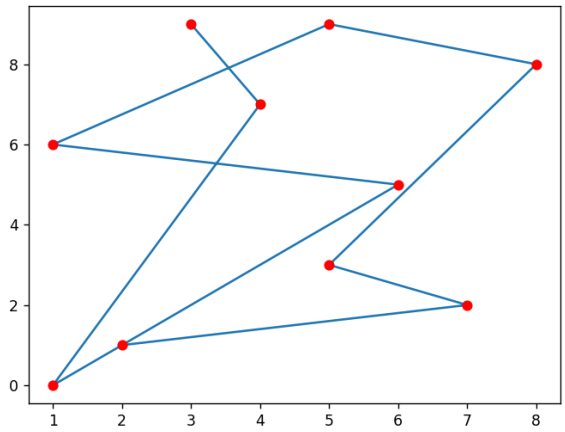
Com estas análises concluímos que na geração 13 o programa conseguiu encontrar o melhor caminho possível, pois acabou por estagnar e devolver todos os indivíduos a mesma solução da geração 13, com apenas 13 gerações foi possível obter o melhor *path* com 10 cidades.

Utilizando a probabilidade 0.75 de mutação e 0.75 de crossover:

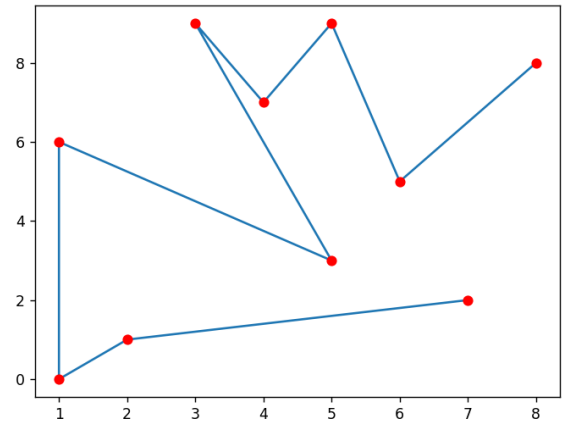
Geração 1, fitness 0.0263264:



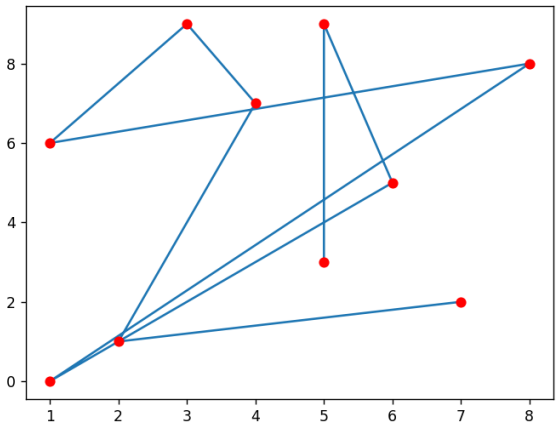
Geração 3, fitness 0.02661603:



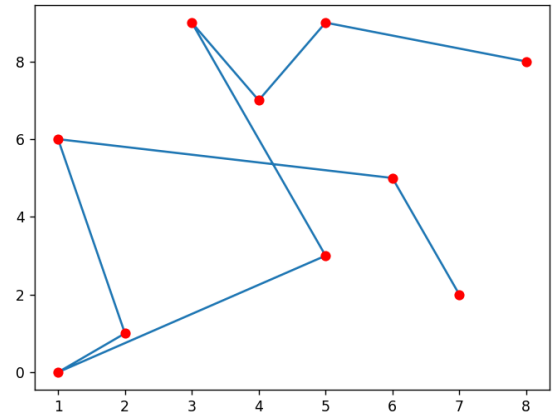
Geração 13, fitness 0.02684586:



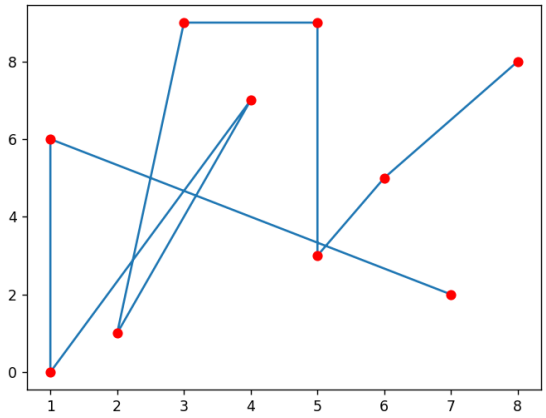
Geração 14, fitness 0.02774804:



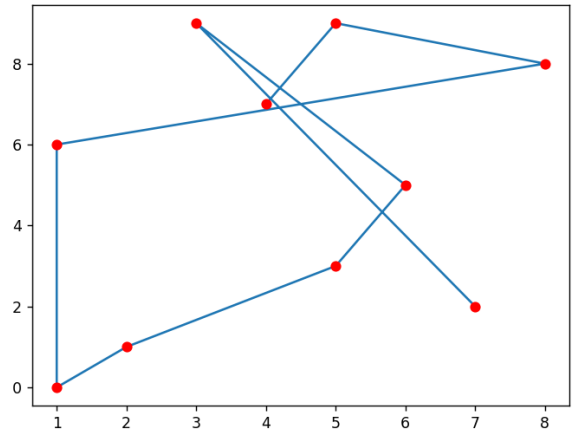
Geração 15, fitness 0.029644123:



Geração 18, fitness 0.031753012:



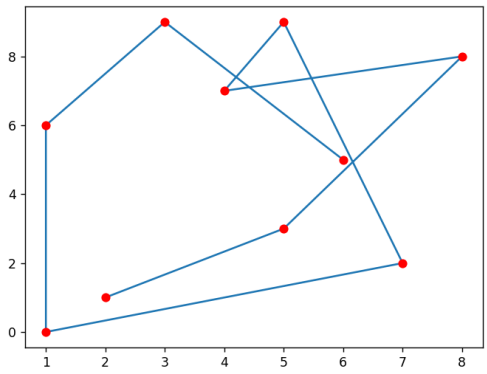
Geração 27, fitness 0.032764297:



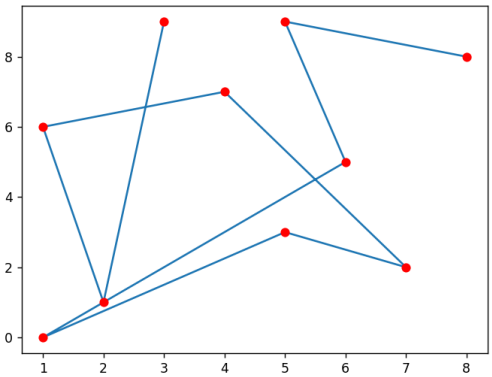
Com estas análises concluímos que na geração 27 o programa conseguiu encontrar o melhor caminho possível pois acabou por estagnar e devolver em todos os indivíduos a mesma solução da geração 27, com apenas 27 gerações foi possível obter o melhor path com 10 cidades.

Utilizando a probabilidade 0.5 de mutação e 0.5 de crossover:

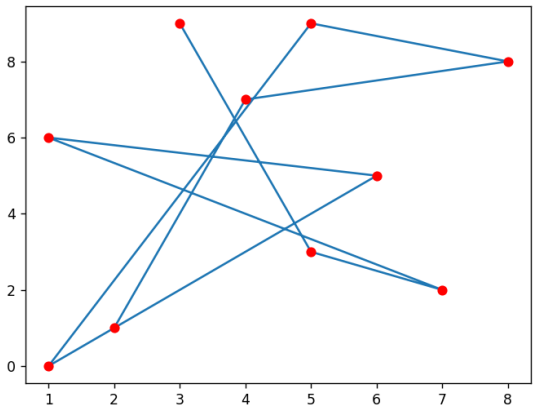
Geração 1, fitness 0.02272422:



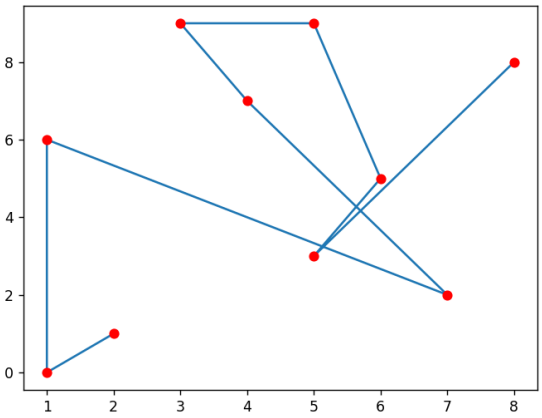
Geração 2, fitness 0.02285869:



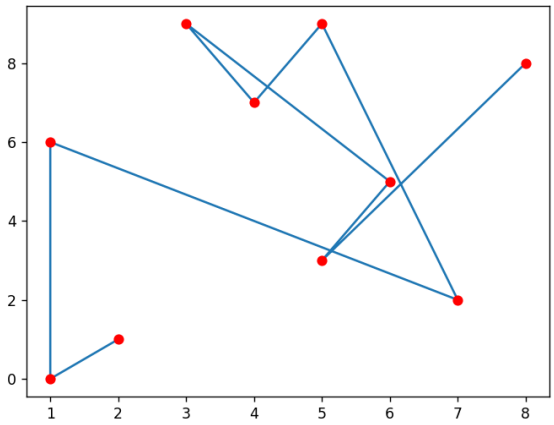
Geração 4, fitness 0.02337753:



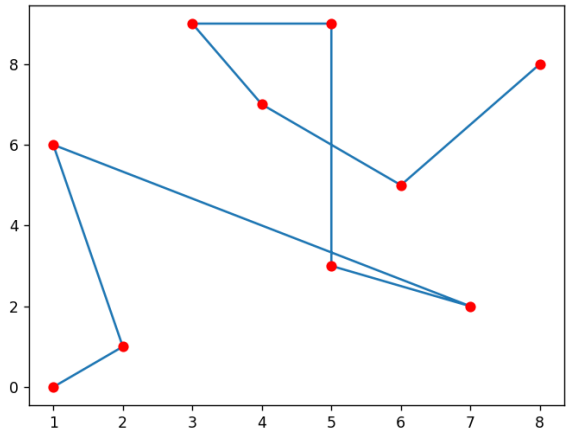
Geração 5, fitness 0.02702147:



Geração 8, fitness 0.027113157:



Geração 14, fitness 0.0306462215:



Com estas análises concluímos que na geração 14 o programa conseguiu encontrar o melhor caminho possível pois acabou por estagnar e devolver em todos os indivíduos a mesma solução da geração 14, com apenas 14 gerações foi possível obter o melhor path com 10 cidades.

0.25 pm e pc:
0.03141102 (max) - 0.0233462 (min) = 0.00806482

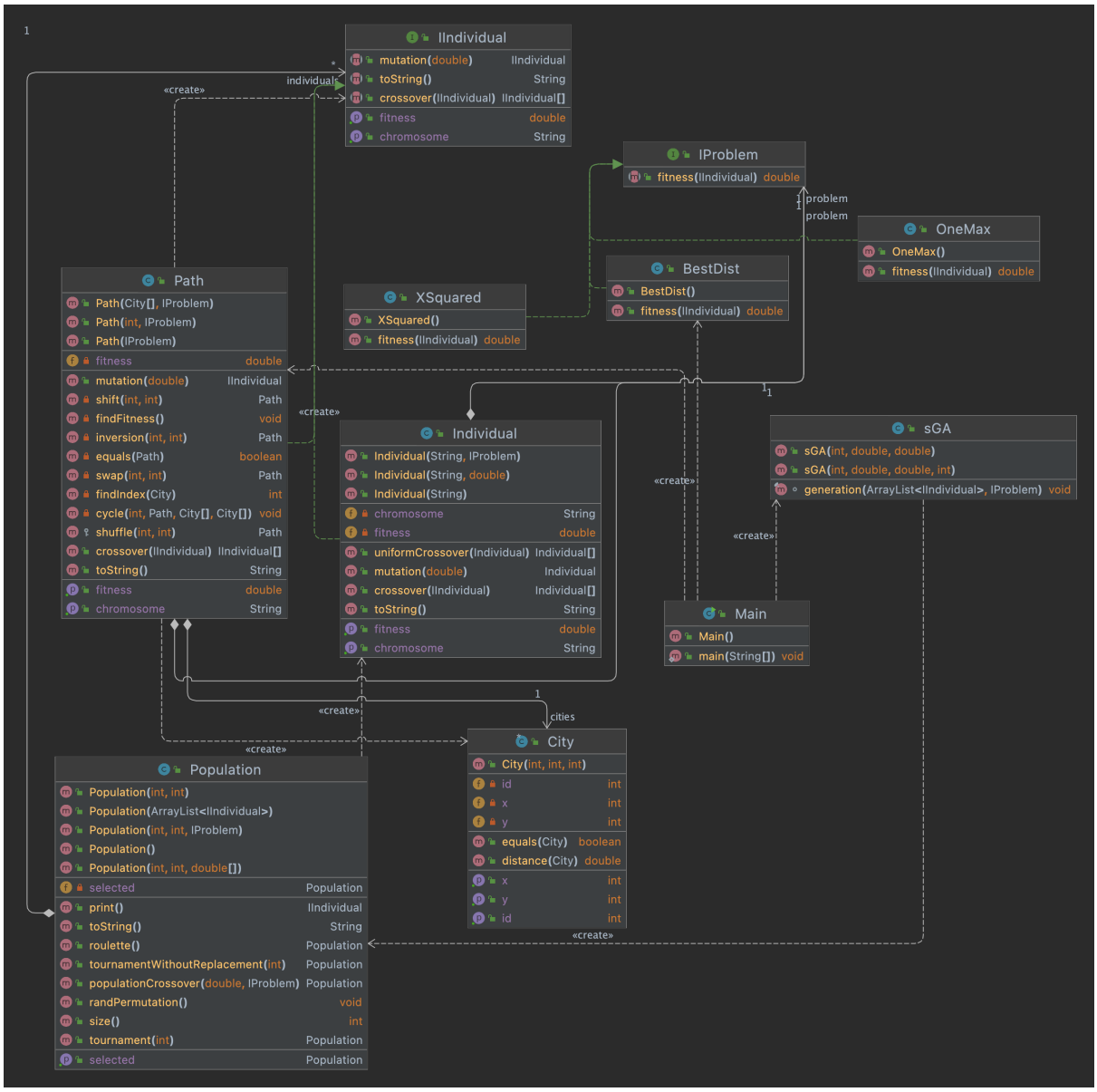
0.75 pm e pc:
0.032764297 (max) - 0.0263264 (min) = 0.006437897

0.5 pm e pc:
0.030646221 (max) - 0.02272422 (min) = 0.007922001

Com base nestas análises concluímos que o melhor desempenho é utilizar as probabilidades 0.25 nas mutações e no crossover, pois para além de conseguir o melhor *path* em menos gerações, apenas 12, conseguiu o maior aumento de fitness.

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 genéticos, 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

Slides disponibilizados pelo docente na tutoria eletrónica.