Inteligência Artificial

Projeto 1

CheckPoint 1

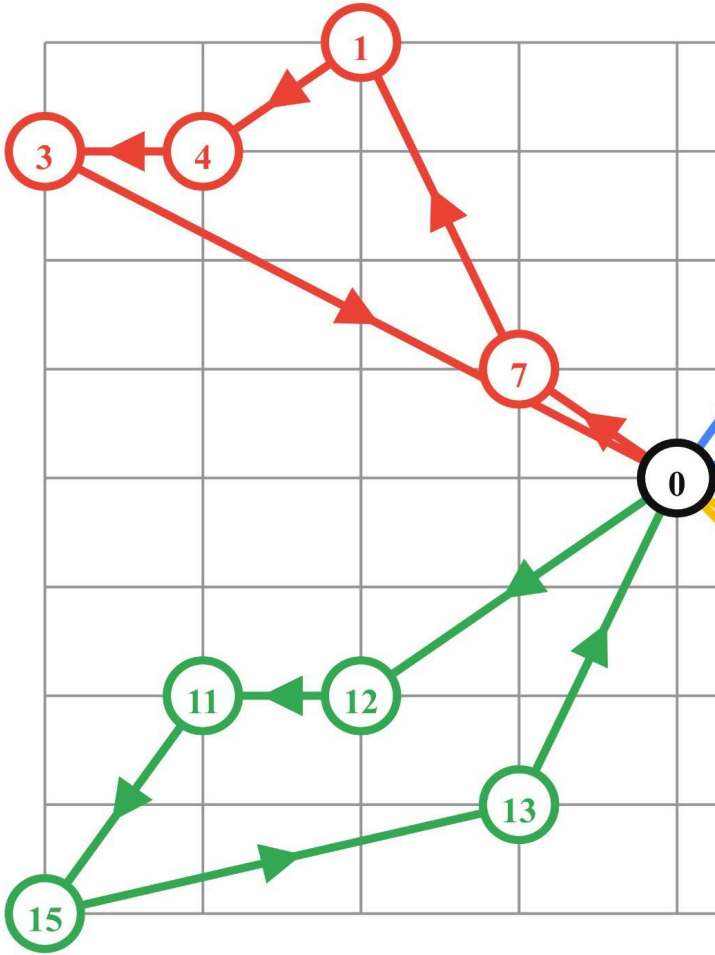
Especificação Problema

3A) Minimização do tempo de viagem

Versão simplificada do problema da ASAE de inspeção de estabelecimentos -> **Vehicle Routing Problem with Time Windows** - com restrição de horários (VRPTW)

Características:

- Conjunto finito de veículos (k) que têm de partir e chegar ao mesmo local (Depot)
  *k = floor(0.1 * nº_estabelecimentos) -> 10%*
- Partida às 9:00 AM de todos os veículos e o tempo de horas de trabalho é ilimitado
- A cada veículo é associada uma rota que contém determinados estabelecimentos de inspeção
- Cada estabelecimento só é inspecionado uma vez
- Cada estabelecimento só pode ser inspecionado se estiver aberto (senão tem de esperar). A partir do momento que o estabelecimento começa a ser inspecionado, a hora a que o estabelecimento fecha é irrelevante
- Objetivo: inspecionar todos os estabelecimentos no mínimo tempo possível (tempo = tempo viagem + espera + inspeção)



Dataset

“establishments.csv”
informação geral de 1000 estabelecimentos (id, nome, inspection time (minutos), horas em que o estabelecimento está aberto (vetor binário com tamanho=24), utilidade (importância para a sociedade), outros dados em princípio inúteis para o nosso problema)

“distances.csv”
distâncias (em segundos) entre todos os estabelecimentos (tabela 1000x1000, excluindo as linhas dos indexes)
Nota: linha (partida) e coluna (chegada) **??**

Output

Solução: para cada veículo, lista de estabelecimentos inspecionados (ID) por ordem

Métricas: tempo total de viagem, tempo total de espera, número de veículos utilizados, valor de utilidade para cada veículo, número de estabelecimentos inspecionados (deverá ser sempre todos)

Métricas de performance: tempo total de execução do algoritmo, tempo até atingir solução ótima, número de iterações até atingir solução ótima, número total de iterações

Bibliografia

Wikipedia

Introdução do problema Vehicle Routing Problem

- Generalização do TSP (Travelling Salesman)
- Solução ótima - NP HARD

(https://en.wikipedia.org/wiki/Vehicle_routing_problem)

Science Direct

Artigos científicos com informação prática e teórica sobre meta-heurísticas no Vehicle Routing Problem

(https://www.sciencedirect.com/topics/economics-econometrics-and-finance/vehicle-routing-problem)

Tese Mestrado Minho

Sobre Vehicle Routing Problem (inclui VRPTW) Informação, maioritariamente teórica, sobre o problema, possíveis restrições, algoritmos greedy, algoritmos exatos (branch and bound) e 3 meta-heurísticas (http://repositorium.sdum.uminho.pt/handle/1822/22289)

Capítulo Springer

Capítulo sobre VRPTW e algoritmos genéticos

(https://link.springer.com/chapter/10.1007/978-981-13-0761-4_52)

Formulação do Problema - *otimização*

Representação solução	Para cada veículo: lista com os estabelecimentos visitados (ID) por ordem
Vizinhança/Mutação/Crossover	Sempre tendo em conta as restrições: 1. Escolher um estabelecimento aleatório e trocá-lo para um veículo diferente (aleatório) 2. Trocar 2 estabelecimentos de diferentes veículos (escolha aleatória) 3. Trocar a ordem de 2 estabelecimentos dentro do mesmo veículo (Ainda pretendemos investigar mais)
Restrições:	• Cada veículo começa e acaba no mesmo sítio (Depot) • Máximo k veículos • Cada estabelecimento só é inspecionado 1 vez • Todos os estabelecimentos são inspecionados
Função avaliação:	1º - Cálculo do tempo total para cada veículo (tempo viagem + espera + inspeção) 2º - Cálculo do máximo desses valores (calculando assim o tempo que demora o carro que tem o trajeto mais longo, ou seja, chega de regresso depois de todos os outros)
Extra - Solução inicial:	Utilizar todos os veículos disponíveis, distribuir aleatoriamente todos os estabelecimentos e cada veículo começa e acaba no Depot

Implementação

Linguagem

Python

Bibliotecas

Pandas: importação de dados de csv

NumPy: processamento de grandes matrizes e funções matemáticas

Matplotlib: criação de gráficos

Meta-heurísticas

- Hill-climbing
- Simulated annealing
- Tabu search (evitar ciclos e soluções recentemente analisadas)
- Genetic algorithms

Visualização

Visualização gráfica (com informação extra em texto) da evolução da qualidade da solução

Interface

Interface em linha de comando para selecionar um algoritmo e os diferentes parâmetros de execução

Nota: uma vez que o problema que escolhemos era de otimização, decidimos esperar pela aula teórica sobre este assunto (2º-feira) para iniciar o trabalho