

Introdução à Inteligência Artificial

Licenciatura em Engenharia Informática, Engenharia Informática – Pós-Laboral e Engenharia Informática – Curso Europeu

2º Ano – 1º semestre

2024/2025

Trabalho Prático nº 2 ***Problema de Otimização***

1. Objetivos

Conceber, implementar e testar métodos de otimização que encontrem soluções de boa qualidade para diferentes instâncias do problema a seguir descrito.

2. Coin Change Problem

Neste problema, dado um conjunto de N moedas diferentes e um valor V que se quer conseguir, pretende-se encontrar uma solução em que o valor V seja alcançado com o menor número de moedas possíveis.

A qualidade da solução é o número de moedas usadas para somar o valor desejado. No caso de soluções onde o valor não seja igual a V (soluções inválidas) a qualidade deve ser penalizada, ou a solução reparada.

Por exemplo, vamos assumir $N = 3$, isto é, que se podem usar 3 moedas diferentes para atingir um valor V . Vamos escolher as moedas de 10 cêntimos, 20 cêntimos, 50 cêntimos. Ou seja, cada moeda tem os valores 0.1, 0.2 e 0.5. Assuma que pretende atingir $V = 1$ euro usando apenas estas moedas.

Há várias soluções possíveis:

- S1 – 10 moedas de 10 cêntimos = 1 euro
- S2 – 5 moedas de 20 cêntimos = 1 euro
- S3 – 5 moedas de 10 cêntimos e uma moeda de 50 cêntimos = 1 euro
- S4 – 3 moedas de 10 cêntimos, 1 moeda de 20 cêntimos e 1 moeda de 50 cêntimos = 1 euro
- S5 – 2 moedas de 50 cêntimos = 1 euro
- e outras

Fazendo uma escolha aleatória das moedas também se podem encontrar soluções inválidas, por exemplo se escolher 5 moedas de 10 cêntimos e duas de 50 cêntimos, soma 1.5 euros que corresponde a uma solução inválida (o V desejado é 1 euro). Este tipo de soluções deve ser tratado pelos algoritmos usando as abordagens dadas nas aulas.

De acordo com o objetivo do problema, das soluções válidas mostradas acima, a solução S5 é a melhor solução, uma vez que é a que usa o menor número de moedas para atingir o valor V .

Instâncias para teste

Estão disponíveis no *Moodle* algumas instâncias deste problema para teste. O formato dos ficheiros de texto, onde está armazenada a informação de cada uma das instâncias, é o seguinte:

- Primeira linha: N e V – numero de moedas e valor a alcançar
- Segunda linha: valores das N moedas

Por exemplo, para o ficheiro **file1.txt** mostrado abaixo, pretende-se obter a soma de 1 euro. As moedas que pode usar para obter essa soma são as moedas de 5 centimos, 20 centimos e 50 centimos

3	1.0	
0.05	0.2	0.5

Pode usar uma representação com números inteiros, num *array* cujo tamanho é o número de moedas possíveis. Por exemplo, imagine esta solução inicial para o algoritmo de pesquisa local, gerada aleatoriamente:

10	0	1
----	---	---

Esta solução indica que foram escolhidas 10 moedas de 0.05 e uma moeda de 0.5.

Trata-se de uma solução válida (a soma $10 \cdot 0.05 + 1 \cdot 0.5$ totaliza o valor desejado de 1.0 euro), logo a qualidade da solução corresponde ao número de moedas (11)

Aplicando operadores de vizinhança, em que pode remover/adicionar moedas será possível em algumas iterações chegar à solução ótima (de qualidade 2) que usa o menor número de moedas possível para atingir 1.0 euro:

0	0	2
---	---	---

Para soluções inválidas sugere-se que utilize mecanismos de penalização ou reparação. A solução seguinte é inválida porque o somatório das moedas usadas não é 1.0 euro mas sim 0.80

2	1	1
---	---	---

Nestes casos, pode atribuir uma **penalização** que permita distinguir esta solução das soluções válidas, ou aplicar um método de **reparação** que a transforme numa solução válida.

3. Métodos de Otimização

Neste trabalho pretende-se que implemente e avalie a capacidade de diferentes algoritmos de otimização para encontrar soluções de boa qualidade para o problema descrito. Sendo assim, no seu estudo deverá implementar os 3 métodos seguintes e efetuar um estudo comparativo aprofundado sobre o desempenho da otimização. No Moodle estão disponibilizados vários ficheiros de instâncias (.....), com complexidades diferentes, para efetuar o estudo experimental dos vários algoritmos.

Os algoritmos a implementar e testar são os seguintes:

1. Algoritmo de pesquisa local (escolha um apenas entre, trepa-colinas, recristalização simulada ou outro, desde que aprovado pelo professor que lhe vai fazer a defesa)
 - Escolha uma representação apropriada para as soluções;
 - Implemente a função de avaliação apropriada ao objetivo do problema descrito;
 - Proponha estratégias para lidar com soluções inválidas (penalização, reparação, ...);
 - Devem ser exploradas pelo menos duas vizinhanças diferentes.
2. Algoritmo evolutivo
 - Use a mesma representação e a mesma função de avaliação que utilizou no algoritmo de pesquisa local;
 - Proponha estratégias para lidar com soluções inválidas (penalização, reparação, ...);

- Devem ser explorados pelo menos dois operadores de recombinação e dois operadores de mutação diferentes;
 - Devem ser explorados dois métodos de seleção diferentes.
3. Método híbrido combinando as duas abordagens anteriores
- Devem ser exploradas pelo menos duas abordagens híbridas diferentes usando os dois algoritmos implementados nos pontos 1 e 2.

4. Estudo experimental

O estudo experimental deve analisar os parâmetros e componentes de cada algoritmo que podem ter influência no seu desempenho.

No algoritmo de pesquisa local deve realizar o estudo experimental variando:

- Número de iterações
- Se optar pela recristalização simulada: valores de T_{min} , T_{max} , $f. arrefecimento$
- Vizinhanças
- Aceitar soluções de custo igual

No algoritmo evolutivo deve realizar o estudo experimental variando:

- Tamanho da população
- Operadores de recombinação / mutação
- Probabilidades do operador de recombinação / mutação
- Métodos de seleção
- Estratégias de penalização vs reparação para lidar com as soluções inválidas.

Abordagens híbridas

- Escolher os melhores algoritmos encontrados no estudo experimental anterior para construir as abordagens híbridas (melhor pesquisa local + melhor evolutivo)
- Comparar as duas abordagens híbridas

No relatório, após o estudo individual de cada algoritmo deve apresentar uma tabela de comparação com os resultados obtidos por:

- Melhor pesquisa local
- Melhor evolutivo
- Abordagem híbrida 1
- Abordagem híbrida 2

As experiências devem ser repetidas pelo menos 10 vezes e as conclusões do estudo devem ser baseadas na comparação dos valores médios.

5. Critérios de avaliação

- Originalidade e correção dos algoritmos implementados:
 - Pesquisa local (10%)
 - Algoritmo evolutivo (20%)
 - Abordagens híbridas (10%)
- Experimentação e análise:
 - Pesquisa local (15%)
 - Algoritmo evolutivo (25%)
 - Abordagens híbridas (10%)

- Documentação e defesa (10%)

6. Normas de realização do trabalho prático

- O trabalho deve ser realizado em grupos de dois alunos. Em casos excepcionais, com a permissão do docente da turma prática a que assiste, o trabalho poderá ser realizado individualmente. Confirme com o docente da turma prática que frequenta se tal situação é possível.
- Os trabalhos serão sujeitos a defesa obrigatória, em data a anunciar.
- Cotação do trabalho: **6 valores**

7. Entrega do Trabalho

- Data limite para entrega: **23h59m do dia 15 de dezembro de 2024**
- Será dada uma penalização de **25%** por cada dia de atraso;
- Deverá ser submetido um ficheiro compactado em formato ZIP, devidamente identificado com os nomes e números dos alunos que realizaram o trabalho. Esse ficheiro deverá conter o seguinte:
 - **Relatório** abordando, pelo menos, os seguintes pontos:
 - Representação usada para o problema, descrição da função de avaliação e objetivo da otimização;
 - Descrição dos algoritmos e/ou das heurísticas utilizadas. Explicar quais as vizinhanças, métodos de seleção e operadores genéticos implementados;
 - Justificação das principais opções tomadas;
 - Resultados dos testes efetuados e respetiva análise. Os resultados a mostrar no relatório devem ser apenas um resumo baseado em tabelas/gráficos apresentando médias de várias repetições e respetivas conclusões. O estudo completo deve ser anexado num ficheiro Excel.
 - **Código fonte** do programa, executável e exemplos para teste;
 - **Ficheiro Excel** com os testes realizados para cada algoritmo.
- O trabalho deve ser entregue através da plataforma *Moodle* até à data limite indicada.
- As defesas serão nas aulas práticas da semana 16 a 20 de dezembro e noutros dias a combinar caso seja necessário. As defesas são obrigatórias e serão agendadas com os professores das aulas práticas.