

Inteligência Artificial 2019/2020

Ficha de Exercícios 4: Otimização/Meta-Heurísticas

4. Otimização/Meta-Heurísticas

4.1 Resolução de Problema de Geração de Horários

Suponha que dispõe de informação relativa aos alunos inscritos a diversas disciplinas optativas de um curso de mestrado/programa doutoral. Cada disciplina tem unicamente uma aula semanal.

Pretende-se construir um horário utilizando unicamente um dado número de “slots” (horas disponíveis) e minimizando o número de incompatibilidades para os alunos (i.e. disciplinas que os alunos não poderão frequentar por estarem sobrepostas no horário). Para tal, a aula de cada disciplina deverá ser marcada num dado slot (de entre os disponíveis).

Slot 1	Slot 3
Slot 2	Slot 4

Suponha o problema exemplo representado em seguida:

Representação em Prolog

```
slots(4).
disciplinas(12).
disciplina(1,[1,2,3,4,5]). %Alunos 1,2,3,4,5 inscritos à disciplina 1
disciplina(2,[6,7,8,9]).
disciplina(3,[10,11,12]).
disciplina(4,[1,2,3,4]).
disciplina(5,[5,6,7,8]).
disciplina(6,[9,10,11,12]).
disciplina(7,[1,2,3,5]).
disciplina(8,[6,7,8]).
disciplina(9,[4,9,10,11,12]).
disciplina(10,[1,2,4,5]).
disciplina(11,[3,6,7,8]).
disciplina(12,[9,10,11,12]). %Alunos 9,10,11,12 inscritos à disciplina 12
```

Representação em Ficheiro de Texto

```
4 12          // Problema com 4 Slots e 12 Disciplinas
1 2 3 4 5     // Alunos 1,2,3,4,5 inscritos à disciplina 1
6 7 8 9
10 11 12
1 2 3 4
5 6 7 8
9 10 11 12
1 2 3 5
6 7 8
4 9 10 11 12
1 2 4 5
3 6 7 8
9 10 11 12    // Alunos 9,10,11,12 inscritos à disciplina 12
```

a) Construa um predicado Prolog (ou função em linguagem à sua escolha) que lhe permita calcular a tabela de incompatibilidades, ou seja, para cada par de disciplinas calcule o número de alunos que estão inscritos a ambas simultaneamente.

b) Neste problema, a representação mais simples para um qualquer horário (atribuição de disciplinas a slots) consiste em associar, a cada disciplina $1..nd$ um slot $1..ns$, onde nd é o número de disciplinas e ns é o número de slots. Para o efeito, podemos utilizar uma lista de nd inteiros, cujos valores são números de 1 a ns . O índice da lista indicará a disciplina respetiva, e o valor representa o slot ao qual a mesma foi atribuída. Por exemplo, na lista $[4,1,2,3,2,4,1,1,2,1,2,3]$ a disciplina 1 foi atribuída ao slot 4, tal como a disciplina 6.

Construa um predicado Prolog (ou função em linguagem à sua escolha) que lhe permita avaliar uma determinada solução, calculando o número total de alunos inscritos em disciplinas sobrepostas.

c) Construa um programa (em linguagem à sua escolha) que permita utilizar os seguintes métodos para encontrar a solução ótima (ou sub-ótima) para este problema:

- Subida da Colina (várias versões)
- Arrefecimento Simulado
- Programação por Restrições

d) Considere as seguintes soluções iniciais e experimente os vários métodos desenvolvidos:

```
sol_inicial([1,1,1,1,1,1,1,1,1,1,1,1]). % Solução com valores iguais
% sol_inicial([1,1,1,2,2,2,3,3,3,4,4,1]). % Quase a solução?
% sol_inicial([1,1,4,2,2,2,3,3,3,4,4,1]). % Mínimo Local?
% sol_inicial([1,2,3,4,1,2,3,4,1,2,3,4]). % E agora? Outro mínimo local?
```

e) Experimente outras soluções iniciais e experimente utilizar uma solução inicial aleatória e o sistema de “Random Restarts”.

f) Compare os programas desenvolvidos a nível da qualidade da solução obtida e do tempo que demoram a obter a solução para os exemplos apresentados.

g) Experimente criar diversas instâncias do problema, com diferentes dimensões (variando o número de disciplinas e slots) e dificuldades (variando os alunos inscritos nas disciplinas).

h) Se o problema fosse encontrar um horário sem qualquer sobreposição, qual o método que lhe parece mais adequado? Comprove experimentalmente a sua conclusão.