

Scape Coronavírus: implementação de um jogo utilizando a técnica de solução de problemas por restrições (*Constraint Satisfaction Problem – CSP*).

O presente documento descreve tecnicamente a aplicação desenvolvida para o projeto final da disciplina de Inteligência Artificial (IA) do Programa de Pós-Graduação em Engenharia Elétrica e Computação (PPGEEC) da Universidade Presbiteriana Mackenzie (UPM).

A aplicação é jogo com o tema Coronavírus. O mesmo foi nomeado “Scape Coronavírus”. A mesma consiste em uma aplicação *web* que pode ser acessada através do link <https://rsg73626.github.io/docs/scape-coronavirus/>. Sua implementação (códigos) pode ser acessada através do link <https://github.com/rsg73626/scape-coronavirus>.

Visto que o presente documento descreve tecnicamente a aplicação criada, não será explicado quais as regras do jogo nem como jogar. No entanto, essas informações podem ser encontradas na opção “Regras e Como Jogar” presente na página da aplicação (cujo endereço na *web* foi disponibilizado logo acima) - <https://rsg73626.github.io/docs/scape-coronavirus/rules.html>. Assim, o objetivo desse documento é descrever as nuances do código desenvolvido bem como das tecnologias utilizadas.

Organização do projeto

Solução de problemas por restrições

Esta seção foi escrita baseada no conteúdo do capítulo VI do livro “*Artificial Intelligence A Modern Approach Third Edition*”, de Stuart Russel e Peter Norving (edição publicada em 2010). Este foi o livro texto da disciplina durante o semestre em que este documento foi escrito (primeiro semestre de 2020).

Como sugere o nome, esta técnica visa encontrar a solução de problemas a partir da criação de um algoritmo que encontre os valores soluções para o problema que satisfaçam todas as restrições (condições obrigatórias) apresentadas. Para tanto, faz-se necessário modelar o problema. Três elementos fundamentais para essa modelagem são:

- as variáveis: paras as quais serão atribuídos valores do domínio até que se encontre o conjunto de valores que não infringe nenhuma restrição;
- o domínio: conjunto contendo os valores que podem ser atribuídos às variáveis;
- e as restrições: condições obrigatórias que devem ser respeitadas pela solução encontrada.

A partir desses três elementos é realizado uma busca utilizando *backtracking* para tentar encontrar a solução do problema.

Um exemplo clássico da literatura é o problema de definir as cores das regiões de um mapa levando em consideração que duas regiões vizinhas não podem ter cores iguais. Veja Figura 1.

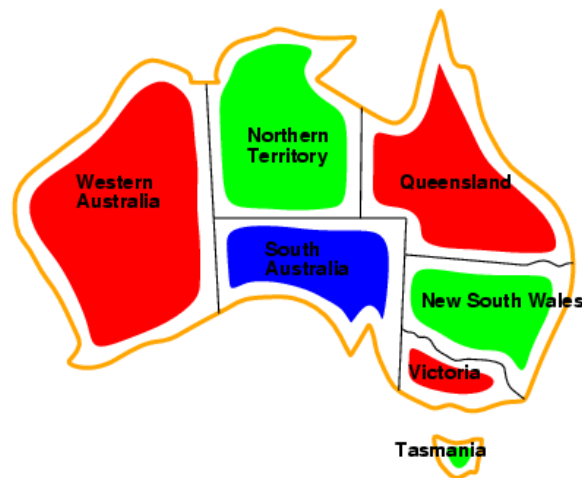


Figura 1. Resolução do problema de pintura de um mapa com restrição de cores diferentes para regiões fronteiriças.

No exemplo da Figura 1 somente três cores foram utilizadas: vermelho, verde e azul. Com isso, para a modelagem de um CSP, estes seriam os valores do elemento **domínio**. Os outros dois elementos são a lista com o nome de cada região, para as **variáveis**, e as ligações entre as regiões que fazem fronteira, para as **restrições** (tais ligações podem ser apresentadas em forma de grafo – veja Figura 2).

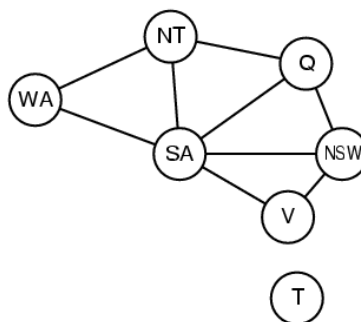


Figura 2. Possível representação em grafo das restrições do problema de pintura das regiões de um mapa com restrição de cores diferentes para regiões fronteiriças.

Algoritmo

Pontos importantes do algoritmo.

Implementação

Grafo

CSP

Exemplo de utilização da classe CSP para resolução do problema da seção anterior

GameBoard

GameView

GameViewController