Elementos de Programação

Projecto de Computação Evolutiva

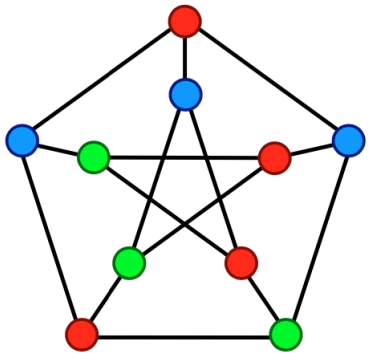
Departamento de Matemática, IST

LMAC
Outubro de 2023

O Problema do Número Cromático

Há muitos exemplos de problemas de grande utilidade prática para os quais não é conhecida uma solução geral eficiente. Um desses problemas consiste em determinar o número cromático de um grafo (não orientado), com aplicações, nomeadamente, em questões de planeamento de tarefas e de reconhecimento de padrões. Uma coloração de um grafo é uma atribuição de cores aos nós do grafo de forma a que quaisquer dois nós que sejam ligados por um arco tenham cores distintas. O número cromático de um grafo é o menor número de cores distintas que são necessárias para o colorir.

Para ilustração, considere-se o grafo representado abaixo, conhecido por grafo de Petersen, com 10 nós e número cromático 3.



Uma instância muito conhecida deste problema geral é abordada pelo famoso *Teorema das Quatro Cores*, que estabelece que o número cromático de qualquer grafo planar (que pode ser desenhado no plano sem cruzamento de arestas) é no máximo 4, ou equivalentemente, que qualquer mapa pode ser colorido com apenas 4 cores de forma a que regiões do mapa separadas por uma linha de fronteira tenham sempre cores distintas. Este resultado foi demonstrado em 1976 por K. Appel e W. Haken, e trata-se do primeiro exemplo relevante de uma demonstração matemática obtida com a ajuda de meios computacionais (usados para tratar cada um dos 1936 casos possíveis identificados).

Apesar de em geral ser fácil verificar se uma dada coloração dos nós de um grafo é ou não uma coloração válida, com generalidade não é conhecido nenhum algoritmo eficiente para calcular o número cromático de um grafo. Na prática, todos os algoritmos conhecidos consistem essencialmente em verificar uma a uma todas as colorações possíveis do grafo. Por exemplo, dado um grafo com 100 nós, para testar todas as suas possíveis colorações com 2 cores num computador com 10GHz (que executa 10^{10} operações por segundo) seria necessário um tempo na ordem de $2^{100}/10^{10}$ segundos, mais de 100 vezes a idade estimada do universo.

Na verdade, este é um problema dito NP-difícil, uma classe de problemas para os quais se conjectura não existir nenhum algoritmo eficiente*. Assim sendo, é usual recorrer-se a soluções aproximadas do problema.

Computação Evolutiva

O objectivo deste projecto é desenvolver em *Python* um programa que calcule uma solução aproximada para o problema do número cromático para um grafo G dado, utilizando o paradigma de computação evolutiva. A ideia consiste em simular durante um período de tempo dado *Tfim* a evolução de uma população inicial formada por um número dado K de indivíduos (que no caso devemos entender como possíveis colorações de G) de acordo com a sua adaptação ao problema (que deve ser maior para uma coloração válida e, entre as colorações válidas, deve ser maior para as colorações que utilizem menos cores). Após a simulação, se houver sobreviventes, a solução aproximada calculada corresponderá ao melhor adaptado dos indivíduos da população final.

O coeficiente de adaptação de um indivíduo com coloração C num certo instante é dado por

$$\begin{cases} \frac{N(G)}{\#(C)} & \text{se } C \text{ é uma coloração válida de } G \\ \\ \frac{1}{1+D(G,C)} & \text{caso contrário} \end{cases},$$

onde $N(G)$ é o número de nós do grafo G, $\#(C)$ é o número de cores distintas utilizadas pela coloração C, e $D(G,C)$ é o número de defeitos da coloração, isto é, o número de arestas de G que, de acordo com a coloração C, unem nós com a mesma cor.

Cada indivíduo da população deve ter um identificador único irrepetível, sendo que poderão coexistir vários indivíduos diferentes correspondentes à mesma distribuição de cores. Assume-se, precisamente, que a população inicial é formada por indivíduos com a mesma coloração uniforme (uma cor distinta para cada nó de G). Ao longo da simulação, cada indivíduo pode sofrer mutações (alterações de coloração), reproduzir-se (dando origem a novos indivíduos com colorações obtidas a partir da do progenitor) ou morrer, de acordo com leis aleatórias que dependem de parâmetros dados *Tritmo*, *Tlimiar* e *Tfiltro*, e também do coeficiente de adaptação do indivíduo, e eventualmente da sua idade. Por razões óbvias, consideram-se apenas colorações obtidas a partir de um máximo de $N(G)$ cores distintas.

O simulador deve ser construído de acordo com a técnica de simulação digital estocástica por sequenciamento de eventos pendentes, havendo que considerar eventos de 3 tipos:

*Há um prémio de um milhão de dólares para quem encontrar um algoritmo eficiente para resolver o problema, ou provar que tal algoritmo não existe (ver a página do prémio atribuído pelo Clay Mathematics Institute).

- *avaliação*, de certo *indivíduo*, cuja cadência é uma variável aleatória exponencial de valor médio *Tlimiar*, e cuja ocorrência resulta na morte do indivíduo, se ainda for vivo nesse instante, com uma probabilidade[†] dada por

$$1 - \frac{2}{\pi} \arctan((1+A)^{1+\frac{8}{1+I}})$$

  onde A é o coeficiente de adaptação do indivíduo nesse instante, e I é a sua idade;

- *evolução*, de certo *indivíduo*, cuja cadência é uma variável aleatória exponencial de valor médio *Tritmo*, e cuja ocorrência resulta, caso ainda esteja vivo nesse instante, numa *mutação* do indivíduo, com probabilidade *Prob*, ou numa *reprodução* do indivíduo, com probabilidade $1-$*Prob*; em caso de *mutação* a coloração do indivíduo altera-se exactamente num nó, escolhido uniformemente, que passa a ter uma cor escolhida, também uniformemente, entre as cores já existentes na sua coloração (esta alteração tem efeito apenas se a nova coloração obtida suscita uma melhoria do coeficiente de adaptação); em caso de *reprodução* é criado um novo indivíduo nesse instante, cuja coloração se obtém como no caso anterior a partir da coloração do progenitor; o valor de *Prob* é dado por

$$\frac{1}{1+e^{\frac{K-T}{10}}}$$

  onde T é o tamanho da população no instante em que ocorre o evento;

- *selecção*, cuja cadência é a constante *Tfiltro*, e cuja ocorrência resulta na morte de todos os indivíduos cuja coloração nesse instante não seja válida, e eventualmente também na morte dos indivíduos remanescentes com pior coeficiente de adaptação, por forma a garantir que a população resultante não exceda $\frac{3}{2}K$ indivíduos.

Objectivos

O projecto deve ser desenvolvido de acordo com o método de programação modular, por camadas, centrado nos dados.

1. Comece por caracterizar os tipos de dados relevantes[‡], nomeadamente *grafos*, *colorações*, *indivíduos*, *população*, *eventos*, e *cadeia de acontecimentos pendentes (CAP)*, e respectivas operações.

2. Desenvolva de seguida o programa abstracto pretendido sobre a camada que disponibiliza estes tipos de dados.

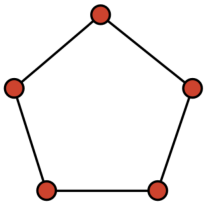
3. Implemente estas camadas sobre a camada básica do *Python*.

[†]Note que em *Python*, recorrendo ao módulo predefinido `random`, a expressão `randrange(n)` toma uniformemente cada um dos valores de `0` a `n-1`, isto é, cada valor tem probabilidade $\frac{1}{n}$. Da mesma forma, a expressão `choice(w)` escolhe uniformemente um dos elementos da lista `w`. Adicionalmente, a expressão `random()` devolve um valor real escolhido uniformemente no intervalo $[0, 1[$, pelo que dada uma probabilidade `p` a expressão `random()<p` é verdadeira com probabilidade `p`.

[‡]Note que, quer as cores, quer os identificadores dos indivíduos, podem ser representados por inteiros positivos.

4. Integre o programa obtido em 2 com os módulos desenvolvidos em 3.

5. Experimente o programa desenvolvido com diversos conjuntos de dados à sua escolha, considerando nomeadamente o grafo de Petersen, mas também grafos cíclicos como o da figura abaixo (no caso, com uma coloração obviamente inválida), ou mesmo grafos sem arestas. Deve escolher dados de dimensão e complexidade que sejam representativas da qualidade da implementação proposta, e discutir os resultados obtidos.



Entrega e Avaliação

O projecto, a ser realizado por grupos de 4 alunos, será entregue através do sistema Fénix, após a inscrição do respectivo grupo, até ao final do dia 28 de Outubro de 2023, **impreterivelmente**. A entrega do trabalho deve consistir de um único arquivo (zip ou rar) contendo o simulador, os módulos desenvolvidos, e um pequeno relatório que descreva sucintamente a solução apresentada, os tipos de dados utilizados e respectivas opções de implementação, e uma discussão dos resultados obtidos. Deve adicionar ao arquivo o relatório de auto-avaliação (disponibilizado juntamente com este enunciado) devidamente preenchido pelos elementos do grupo.

O trabalho vale 10 valores da nota final da disciplina, que se distribuem da seguinte forma: descrição dos tipos de dados relevantes e suas operações (2.5 valores), implementação eficiente dos tipos de dados e sua apresentação sob a forma de módulos (3 valores), desenvolvimento do simulador (3 valores), experimentação e discussão dos resultados (1.5 valores).