

Teoria da complexidade e as classes de problemas P, NP e NP-completo

Maria Carolina Resende Jaudacy¹

¹Segundo período do curso de Ciência da Computação
Pontifícia Universidade Católica de Minas Gerais (PUC MG) - Belo Horizonte – MG

mariacarolina_bh@hotmail.com

Alan Turing, importante nome da computação, em 1937 introduziu no cenário a primeira máquina capaz de tratar problemas que poderiam ser, a partir daí, intitulados como "computacionais". Com isso os algoritmos emergiram como formas de solucionar algum problema, sendo este passível de análise sobre ser ou não tratável e qual o nível de complexidade envolvido na implementação de uma solução, em termos computacionais.

O estudo da chamada *medida de complexidade* de um algoritmo (termo criado por Juris Hartmanis e Richard Stearns em um artigo publicado em 1965) surgiu como uma forma de medir e comparar o tempo de execução em máquinas de Turing. A classificação dessa medida veio depois, com as classes de problemas definidas por Jack Edmonds, Richard Karp, Michael Garey, David Johnson e Stephen Cook.

As classes de problemas podem ser entendidas como a classificação dos problemas a serem tratados quanto ao seu nível de solucionabilidade. Elas são conhecidas de modo geral por P (problemas que possuem soluções encontradas de modo eficiente), NP (problemas com soluções verificadas de modo eficiente) e NP-Completo (subclasse de NP, definindo problemas que poderiam ser solucionados em um tempo polinomial P).

Os chamados "problemas de classe P" (sendo P a representação de *Tempo Polinomial Determinístico*) são aqueles que podem ser resolvidos em computadores reais em tempo polinomial, com tempo de execução $O(n^m)$ – ou seja, seu pior caso é determinado por uma função polinomial. Dentre os problemas P encontramos, por exemplo, a verificação da condição de um número como primo e outras questões que envolvem decisão.

Já na classe de problemas NP (*Tempo Polinomial Não-Determinístico*) estão envolvidos todos os problemas que possuem verificação de sua validade realizável em tempo polinomial, mas não resolutivos nesse mesmo tempo. Uma ilustração dessa condição pode ser vista em um jogo de *Sudoku*: a validade de uma solução feita para uma linha qualquer do jogo é passível de ser determinada rapidamente, mas a produção dessa solução, não. É importante deixar claro que todos os problemas de classe P estão inclusos em NP, pois também possuem verificação em tempo polinomial.

Ainda se tratando da classe de problemas NP, encontramos nela a subclasse NP-completo. Nela estão inclusos os problemas que são passíveis de resolução a partir da redução do problema maior em outros pequenos problemas mais simples e com resposta igualmente aplicável à ele, de classe P. Um dos primeiros problemas classificados como NP-completo foi o "Problema de Satisfatibilidade Booleana", popularmente conhecida como SAT, onde seu objetivo é encontrar uma solução verdadeira dada determinada expressão booleana, que se encontrada é considerada satisfatível.

A partir da classificação de problemas em NP-completo surge a discussão sobre um problema P ser ou não NP: se um problema NP não consegue ter sua resolução encontrada em tempo P , nenhum dos problemas NP-completo pode ser resolvido em tempo P , e da mesma forma ocorre o contrário: se algum problema NP-completo consegue ser resolvido em tempo P , todos os problemas NP poderão ser resolvidos em tempo polinomial P .

References

- Cid Carvalho de Souza (2012). Teoria da Complexidade. <http://www.ic.unicamp.br/~zanoni/mc548/2013-2s/material/np.pdf>. Acessado em 21 nov. 2019.
- Igor Carboni Oliveira (2010). Complexidade Computacional e o Problema P vs NP. http://taurus.unicamp.br/bitstream/REPOSIP/275804/1/Oliveira_IgorCarboni_M.pdf. Acessado em 21 nov. 2019.
- Leonardo Ferreira (2016). O problema da satisfatibilidade booleana (SAT). https://ferreirasc.github.io/post/spoj_cardapio/. Acessado em 21 nov. 2019.
- Nair Maria Maia de Abreu (1987). A Teoria da Complexidade Computacional. http://rmct.ime.eb.br/arquivos/RMCT_1_tri_1987/teoria_complex_comput.pdf. Acessado em 21 nov. 2019.
- Paulo Feofiloff (2006). Complexidade computacional e problemas NP-completos. https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/NPcompleto2.html. Acessado em 21 nov. 2019.