

O Problema do Milênio sobre Intratabilidade Computacional

Celina Miraglia Herrera de Figueiredo



Teoria da Computação

Ciência fundamental, assim como Biologia e Física

Por que alguns problemas são fáceis e outros difíceis?

Não estuda quão rápido os computadores são

Astronomia não é o estudo dos telescópios

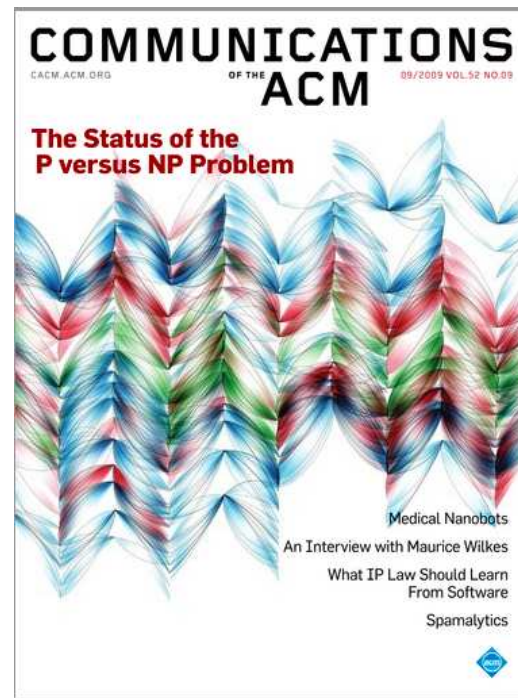
Estuda a estrutura matemática dos problemas

como a estrutura ajuda a resolver ou impede a tentativa de resolver

O Problema do Milênio

Problema central em Teoria da Computação: P versus NP

Existe pergunta cuja resposta pode ser verificada rapidamente, mas cuja resposta requer muito tempo para ser encontrada?



setembro 2009

RESOLVER OU VERIFICAR?

UMA PERGUNTA QUE VALE UM MILHÃO DE DÓLARES

O problema P versus NP: Resolver ou Verificar?

Em 1903, o matemático americano Frank Cole provou que o número

$$2^{67} - 1 = 147573952589676412927$$

não é primo exibindo a fatoração $193707721 \times 761838257287$.

É simples (embora tedioso se feito manualmente) calcular $2^{67} - 1$, calcular o produto $193707721 \times 761838257287$ e verificar que dão o mesmo número. Já encontrar essa fatoração é difícil. Cole disse que ele levou três anos trabalhando aos domingos.

Resolver ou Verificar? é uma pergunta que vale um milhão de dólares

Clay Mathematics Institute Millennium problems



Clay Mathematics Institute

Dedicated to increasing and disseminating mathematical knowledge

[HOME](#) | [ABOUT CMI](#) | [PROGRAMS](#) | [NEWS & EVENTS](#) | [AWARDS](#) | [SCHOLARS](#)

The Millennium Prize Problems

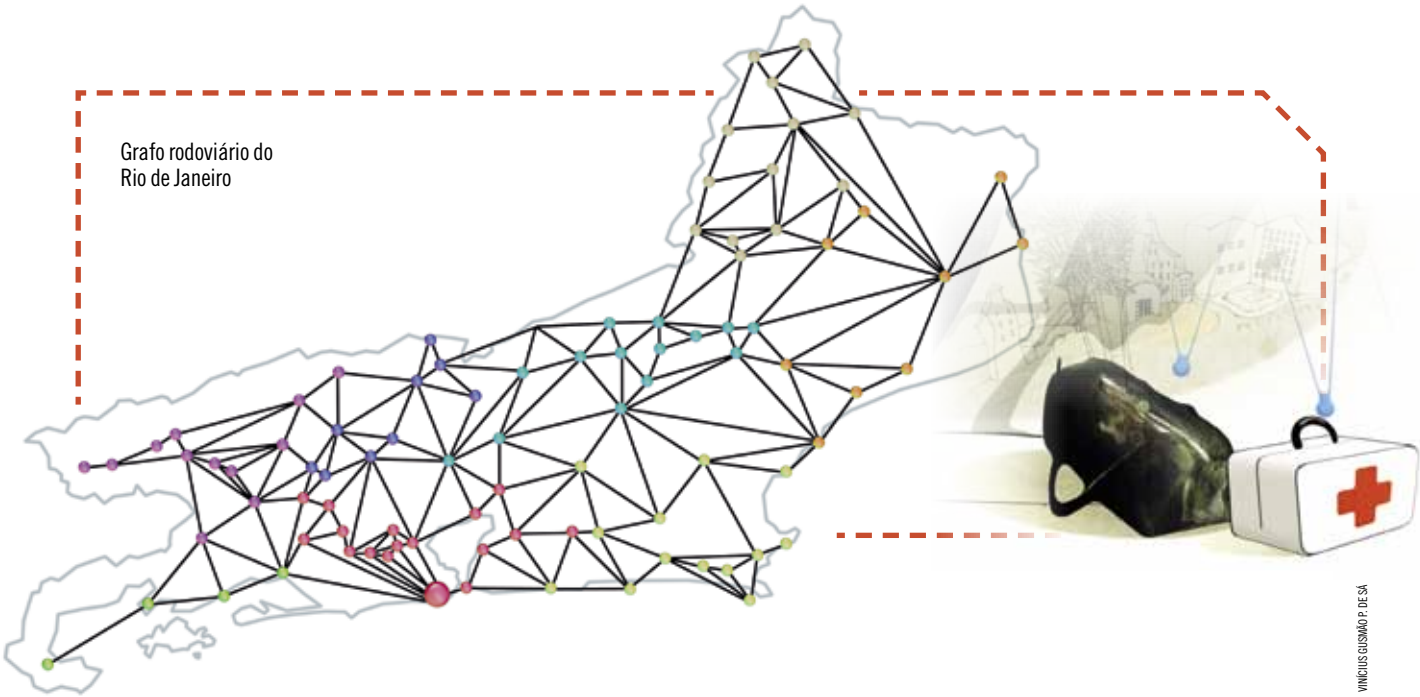
In order to celebrate mathematics in the new millennium, The Clay Mathematics Institute of Cambridge, Massachusetts (CMI) established seven *Prize Problems*. The Prizes were conceived to record some of the most difficult problems with which mathematicians were grappling at the turn of the second millennium; to elevate in the consciousness of the general public the fact that in mathematics, the frontier is still open and abounds in important unsolved problems; to emphasize the importance of working towards a solution of the deepest, most difficult problems; and to recognize achievement in mathematics of historical magnitude.

P vs NP Problem



If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit (by car), how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily (given the methods I know) find a solution.

Grafo rodoviário do Rio de Janeiro



Teoria dos Grafos

A matemática da conectividade, um dos ramos da matemática discreta

O artigo de Euler de 1736 é o nascimento da Teoria dos Grafos

O circuito de Euler corresponde ao percurso do Pavimentador

O ciclo de Hamilton corresponde ao percurso do Vacinador

Euler em seu artigo sobre as pontes de Königsberg estuda como a dificuldade do problema cresce ou escala em função do número de pontes

Euler apresenta uma caracterização

prova de que a cidade admite o circuito de Euler

prova de que a cidade não admite o circuito de Euler

Complexidade computacional

A maioria dos problemas computacionais pertence à classe NP, admitem um certificado polinomial

Em várias e diferentes áreas, procuramos objetos matemáticos: percurso de um caixeiro viajante, atribuição de verdade, emparelhamento máximo, coloração mínima de um grafo

O objeto matemático procurado é o certificado, a prova de que o problema está em NP

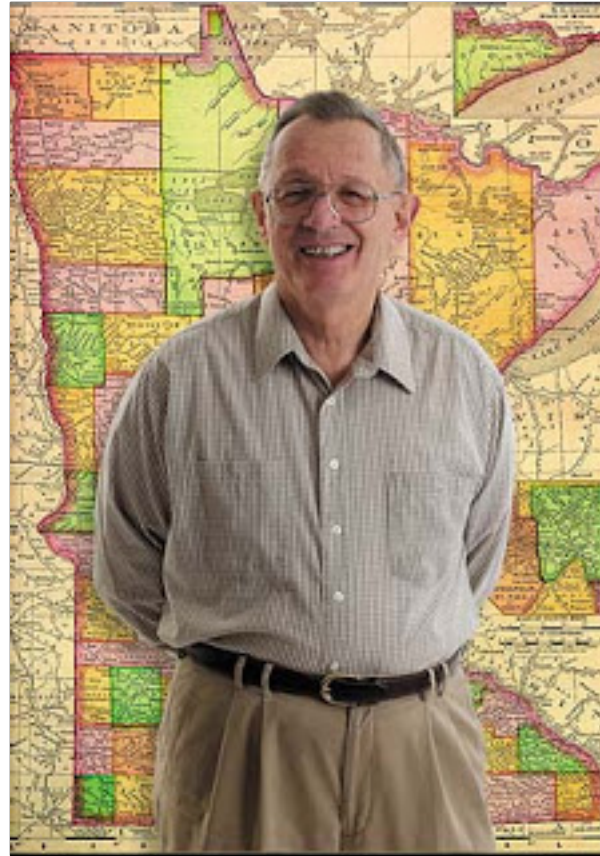
O estudo da complexidade computacional de problemas considera principalmente problemas em NP e tenta distinguir os solúveis em tempo polinomial dos não através da classe dos problemas NP-completos

P vs NP Problem



Suppose that you are organizing housing accommodations for a group of four hundred university students. Space is limited and only one hundred of the students will receive places in the dormitory. To complicate matters, the Dean has provided you with a list of pairs of incompatible students, and requested that no pair from this list appear in your final choice. This is an example of what computer scientists call an NP-problem, since

it is easy to check if a given choice of one hundred students proposed by a coworker is satisfactory (i.e., no pair taken from your coworker's list also appears on the list from the Dean's office), however the task of generating such a list from scratch seems to be so hard as to be completely impractical. Indeed, the total number of ways of choosing one hundred students from the four hundred applicants is greater than the number of atoms in the known universe! Thus no future civilization could ever hope to build a supercomputer capable of solving the problem by brute force; that is, by checking every possible combination of 100 students. However, this apparent difficulty may only reflect the lack of ingenuity of your programmer. In fact, one of the outstanding problems in computer science is determining whether questions exist whose answer can be quickly checked, but which require an impossibly long time to solve by any direct procedure. Problems like the one listed above certainly seem to be of this kind, but so far no one has managed to prove that any of them really are so hard as they appear, i.e., that there really is no feasible way to generate an answer with the help of a computer. Stephen Cook and Leonid Levin formulated the P (i.e., easy to find) versus NP (i.e., easy to check) problem independently in 1971.

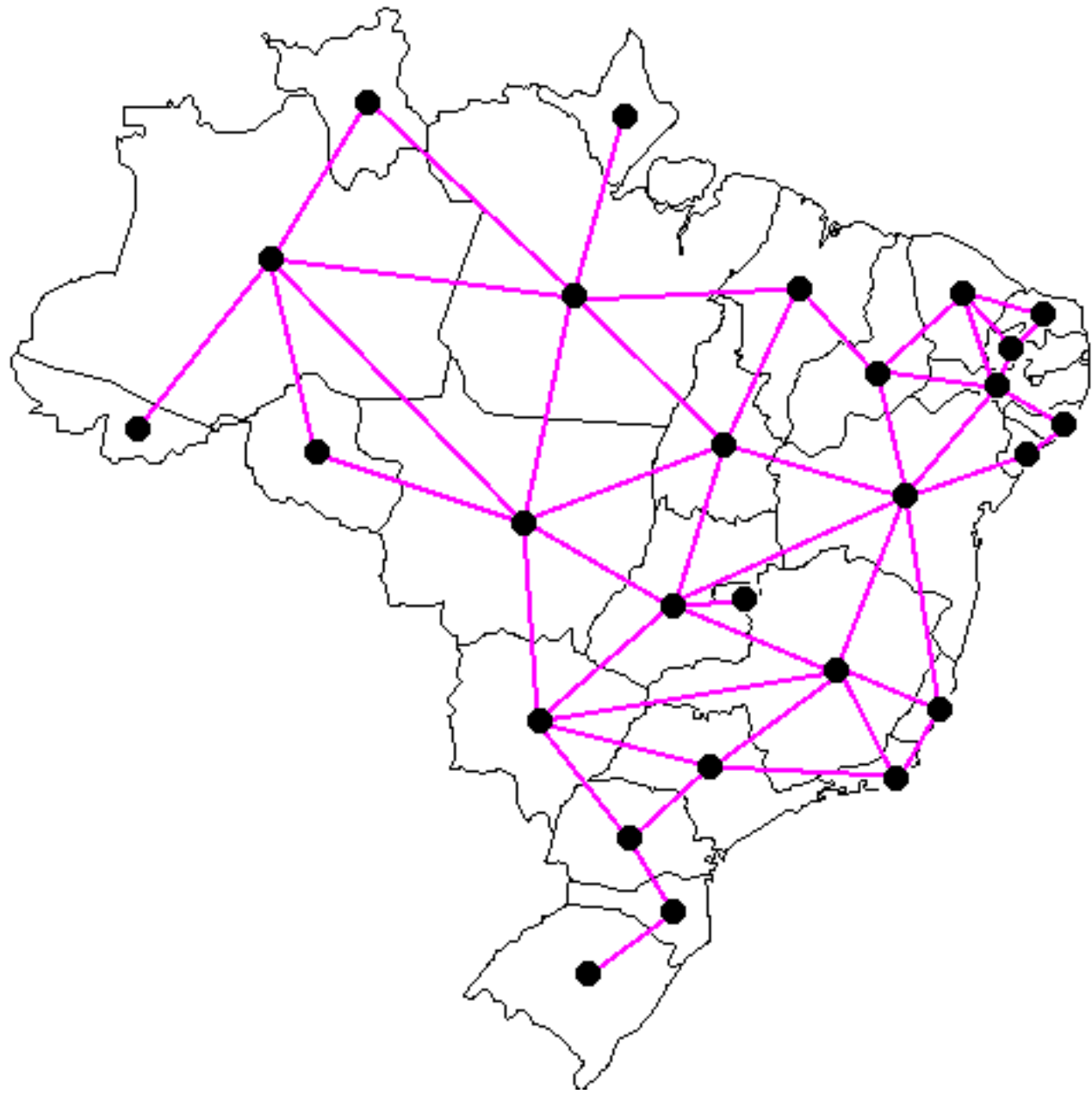


Kenneth Appel

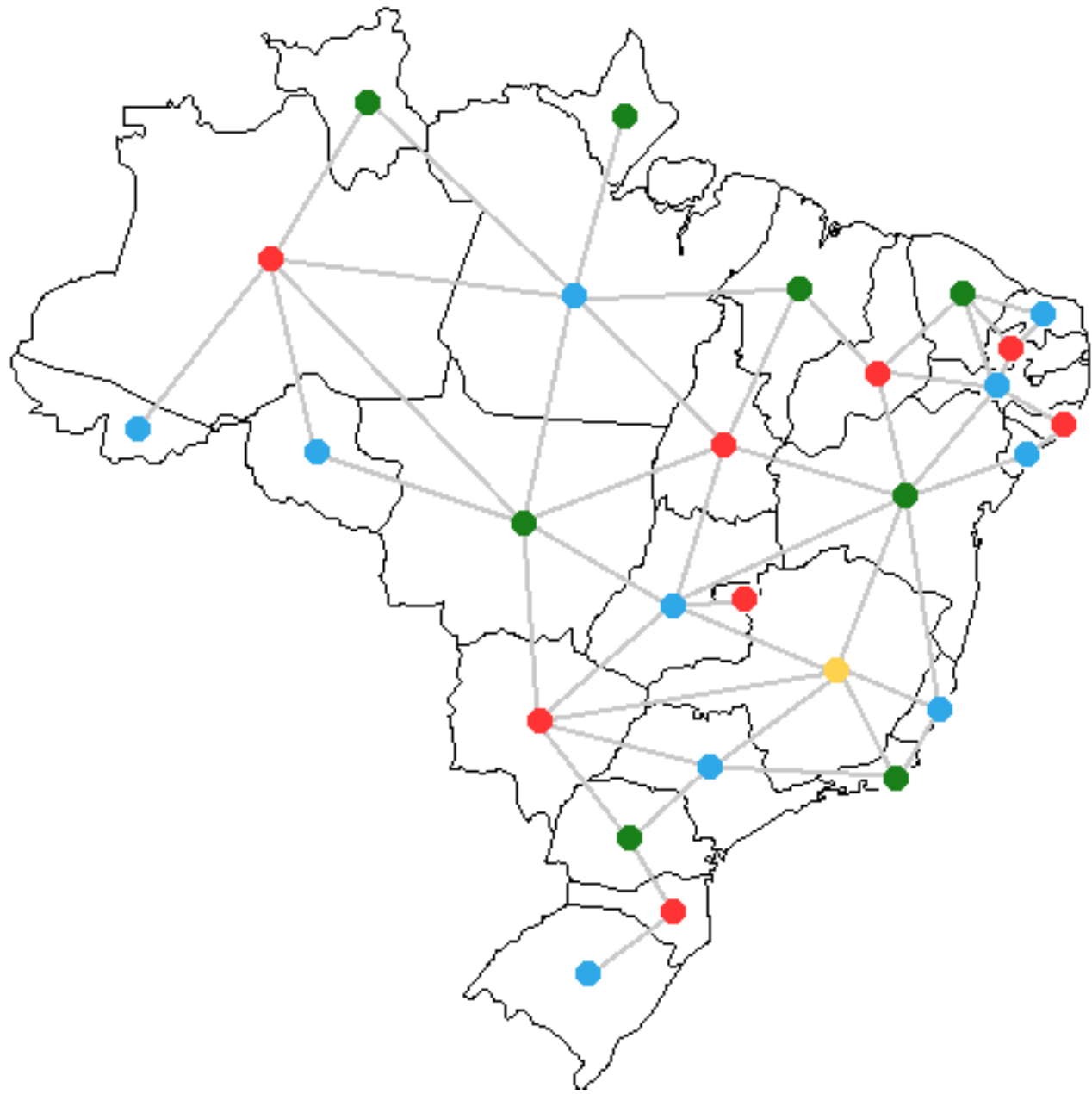
1932–2013













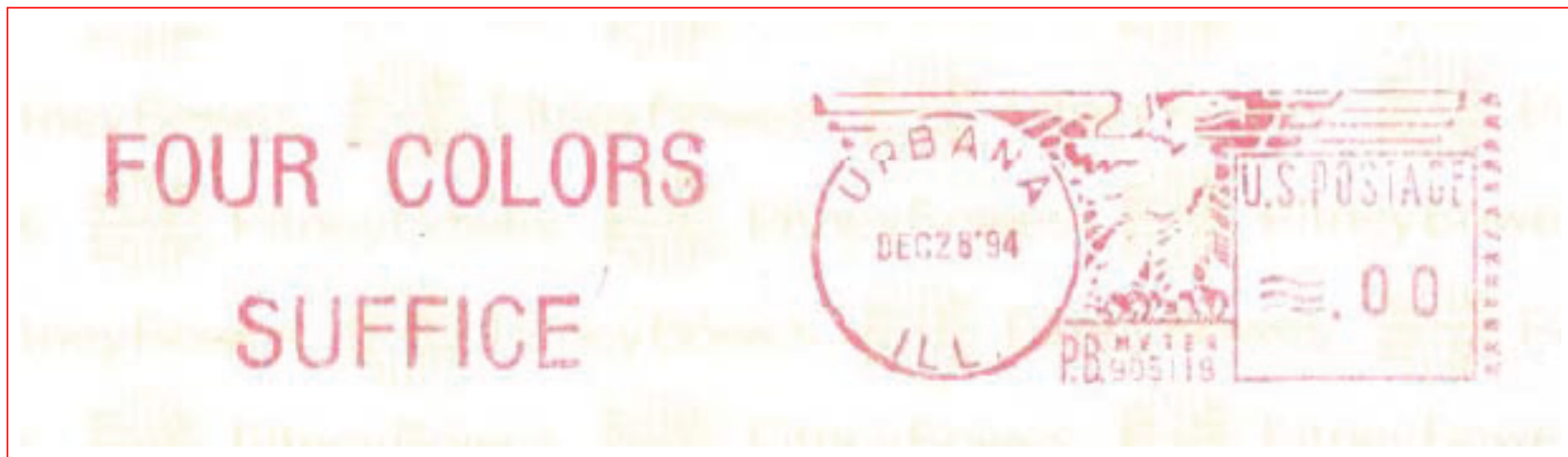
O Teorema das Quatro Cores

Appel e Haken (1977):

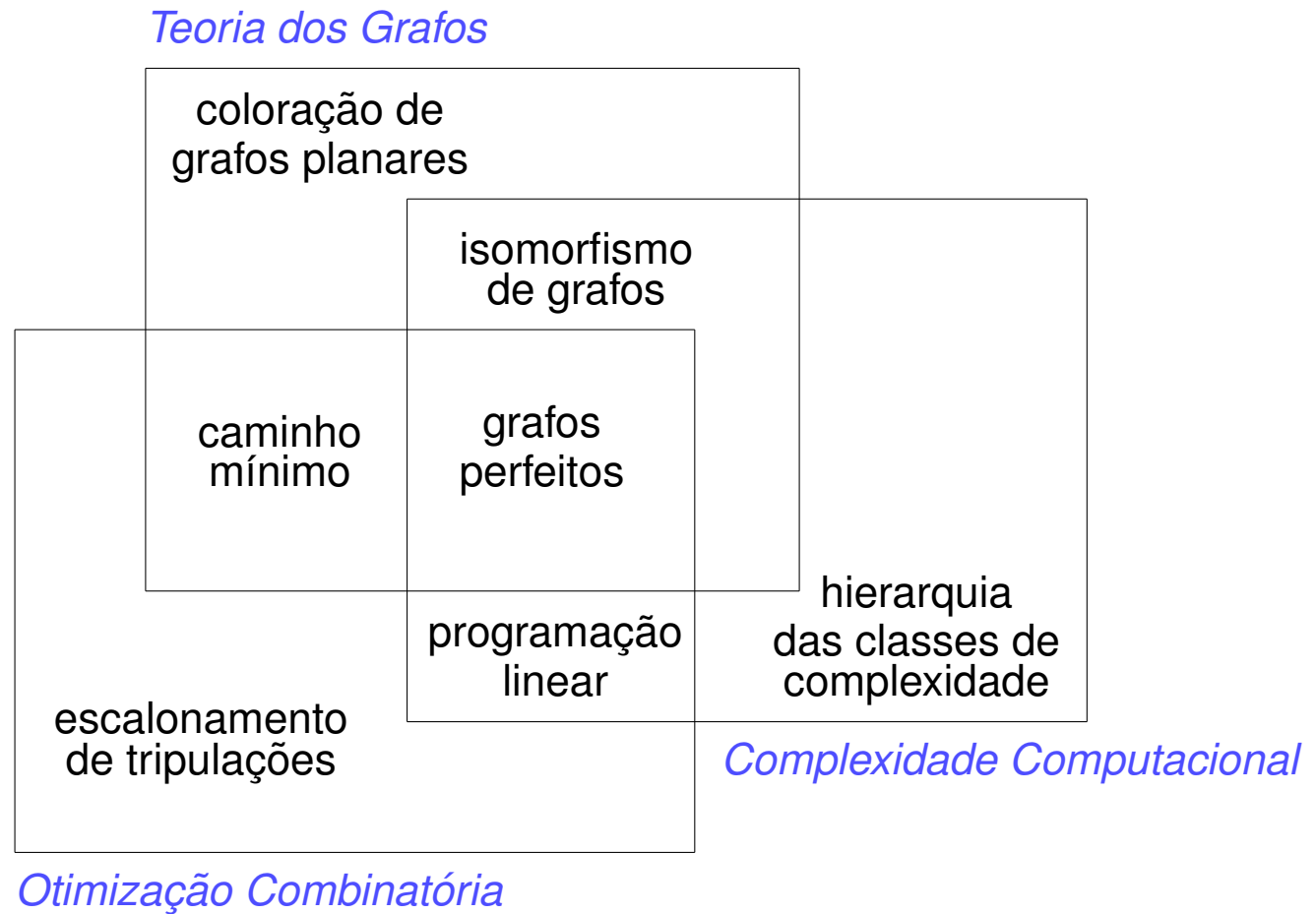
Sim! (prova usando computador)

conjunto inevitável de 1476 configurações redutíveis

algoritmo colore grafo planar com quatro cores em tempo $O(n^4)$



Origem e desenvolvimento da área de pesquisa



Guia de NP-completude

Identificação de problema interessante, de classe de grafos interessante

Classificação da complexidade de um problema: P ou NP-completo

Problema que separa classes de grafos

Classe de grafos que separa problemas

D. Johnson – *J. Algorithms* 1985, *ACM Trans. Algorithms* 2005

M. Golumbic, H. Kaplan, R. Shamir – *J. Algorithms* 1995

J. Spinrad – *Efficient Graph Representations* 2003

Problemas separadores e Classes de grafos separadoras

	VERTEXCOL	EDGECOL	MAXCUT
perfect	P	N	N
chordal	P	O	N
split	P	O	N
strongly chordal	P	O	O
comparability	P	N	O
bipartite	P	P	P
permutation	P	O	O
cographs	P	O	P
indifference	P	O	O
split-indifference	P	P	P

N: NP-completo P: polinomial O: aberto

D. Johnson – *J. Algorithms* 1985

J. Spinrad – *Efficient Graph Representations* 2003

Dicotomias cheias

Classificar todo membro de uma família de problemas como **Polinomial** (fácil) ou **NP-completo** (difícil)

Quais grafos tornam o problema fácil ou difícil?

Teoremas de dicotomia fornecem bons projetos de pesquisa:
simples de formular, mas trabalhoso para concluir

Requerem domínio de técnicas para desenvolvimento de algoritmos e para provas de dificuldade computacional

Grafos sem corda única

	VERTEXCOL	EDGECOL	TOTALCOL
unichord-free	P	N	N
{4-hole,unichord}-free, $\Delta \geq 4$	P	P	P
{4-hole,unichord}-free, $\Delta = 3$	P	N	P

Dicotomia cheia para EDGECOL:

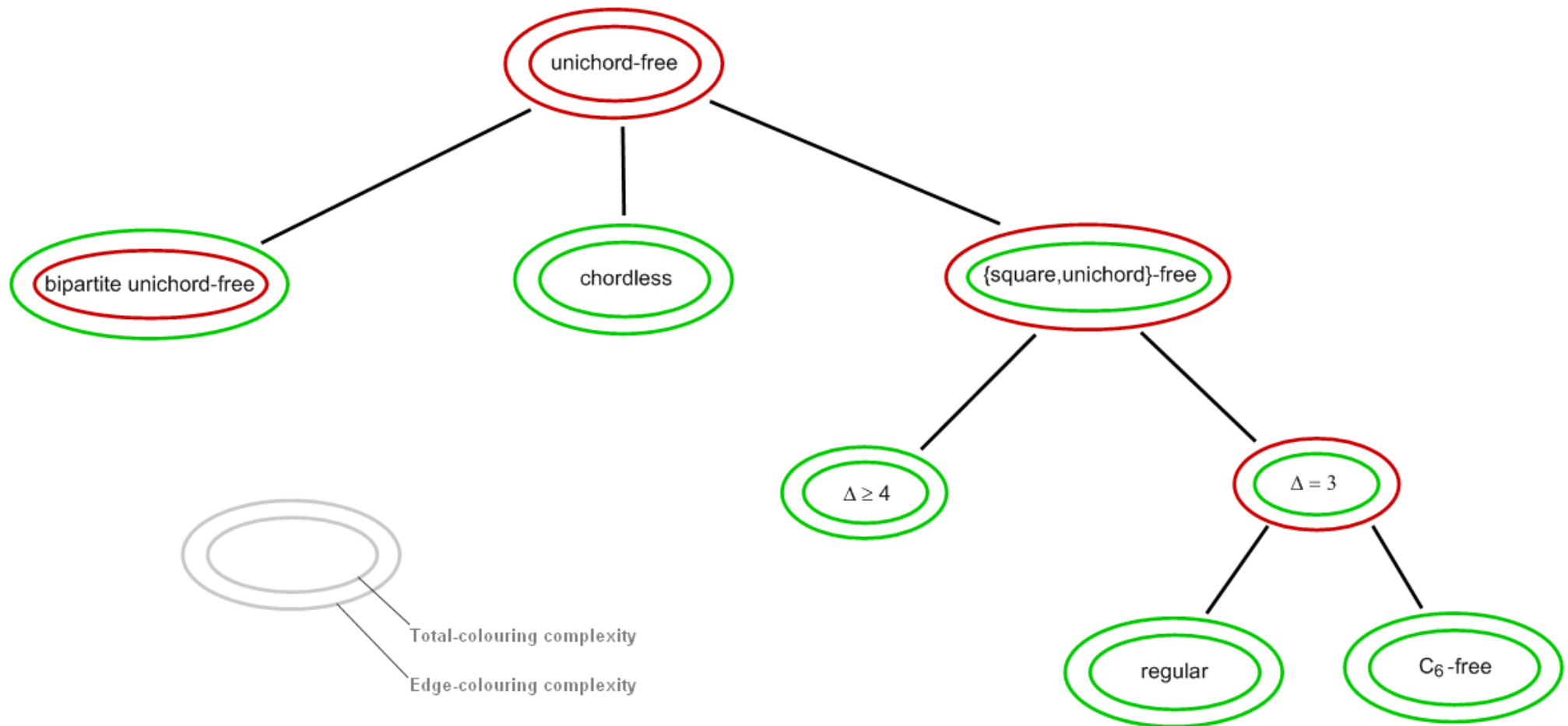
$\Delta \geq 4$ é polinomial mas $\Delta = 3$ é NP-completo

Classe de grafos separadora:

EDGECOL é NP-completo mas TOTALCOL é polinomial

“Complexity of colouring problems restricted to unichord-free and {square,unichord}-free graphs”, *Discrete Appl. Math.* 2014 (with Raphael Machado and Nicolas Trotignon)

Classes de grafos separadoras

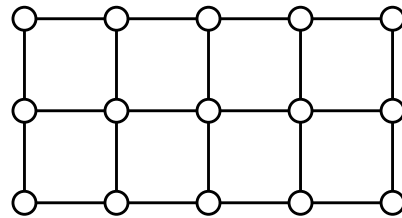


Teoremas de dicotomia podem revelar resultados algorítmicos surpreendentes

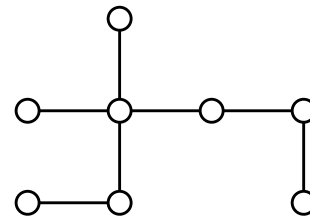
Imersão em grades

Teoria dos grafos: Reconhecimento das grades parciais é problema aberto

Desenho de grafos: Decidir se um grafo admite um VLSI layout com arestas unitárias é NP-completo



a grade $G_{3,5}$



imersão para $\{1,2,4\}$ -árvore

A. Brandstädt, V.B. Le, et al. – Information system on graph class inclusions.

<http://www.teo.informatik.uni-rostock.de/isgci/>, 2002, atualizado 2010

S. N. Bhatt, S. S. Cosmadakis – *Inform. Process. Lett.* 1987

P vs. N dicotomia para grades parciais restritas por graus

D	D-grafos	D-árvores
{1}	P	P
{2}	P	—
{3}	P	—
{4}	P	—
{1,2}	P	P
{1,3}	N	N
{1,4}	P	P
{2,3}	N	—

D	D-grafos	D-árvores
{2,4}	N	—
{3,4}	P	—
{1,2,3}	N [G89]	N [G89]
{1,2,4}	N [BC87]	N [BC87]
{1,3,4}	N	N
{2,3,4}	N	—
{1,2,3,4}	N [BC87]	N [BC87]

Não há conjunto D que defina problema separador

S. N. Bhatt, S. S. Cosmadakis – *Inform. Process. Lett.* 1987

A. Gregori – *Inform. Process. Lett.* 1989

“Complexity dichotomy on partial grid recognition”

Theoret. Comput. Sci. 2011 (com Vinícius Sá, Guilherme Fonseca, Raphael Machado)

Conclusão

Todo grafo é fácil ou difícil

O problema do milênio sobre intratabilidade investiga limitações computacionais fundamentais

Apresentamos dicotomias onde dizemos precisamente quais grafos tornam o problema fácil ou difícil

Exploramos a fronteira de complexidade através da classificação em polinomial ou NP-completo

Bibliografia

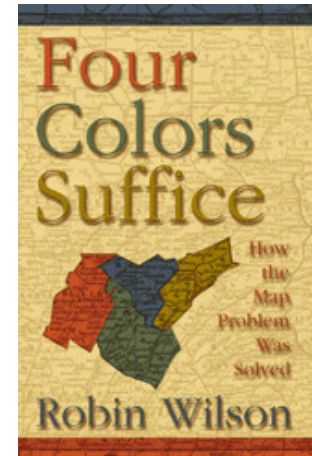
Graph Coloring Problems

Tommy R. Jensen, Bjarne Toft. Wiley, 1995

<http://www.imada.sdu.dk/Research/Graphcol/>

Four Colors Suffice. How the map problem was solved

Robin Wilson. Princeton University Press, 2002



Coloração em Grafos

Celina M. Herrera de Figueiredo, João Meidanis, Célia Picinin de Mello.
XVI JAI, SBC, 1997

<http://www.cos.ufrj.br/~celina/ftp/jai97.pdf>

The P vs. NP-complete dichotomy of some challenging problems in graph theory, Discrete Applied Mathematics (2012)