O Curioso Caso do Problema da Clique Máxima

da série: "A Vida Como Ela É . . ."

Renato Carmo Alexandre Züge . . .

Departamento de Informática da UFPR

12 de agosto de 2021

“A Vida Como Ela É . . .”

“A Vida Como Ela É . . .”

O que fazer quando é preciso resolver instâncias concretas de problemas difíceis?

“A Vida Como Ela É . . .”

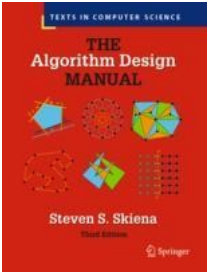
O que fazer quando é preciso resolver instâncias concretas de problemas difíceis?

não existe manual com respostas prontas

“A Vida Como Ela É . . .”

O que fazer quando é preciso resolver instâncias concretas de problemas difíceis?

não existe manual com respostas prontas?

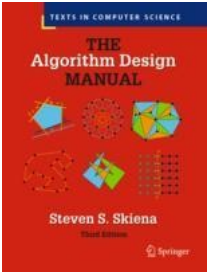


Skiena [2020]

“A Vida Como Ela É . . .”

O que fazer quando é preciso resolver instâncias concretas de problemas difíceis?

não existe manual com respostas prontas?



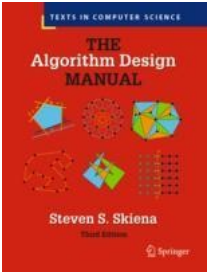
Skiena [2020]

DIMACS Implementation Challenges

“A Vida Como Ela É . . .”

O que fazer quando é preciso resolver instâncias concretas de problemas difíceis?

não existe manual com respostas prontas?



Skiena [2020]

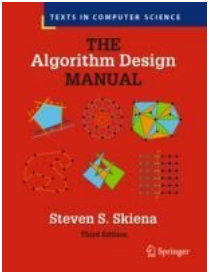
DIMACS Implementation Challenges

> *The Challenges aid in determining realistic algorithm performance where worst-case analysis is overly pessimistic and probabilistic models are too unrealistic*

“A Vida Como Ela É . . .”

O que fazer quando é preciso resolver instâncias concretas de problemas difíceis?

não existe manual com respostas prontas?



Skiena [2020]

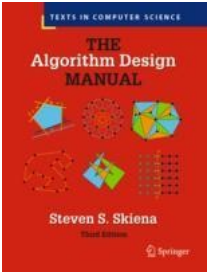
DIMACS Implementation Challenges

> *The Challenges aid in determining realistic algorithm performance where worst-case analysis is overly pessimistic and probabilistic models are too unrealistic Experimentation can provide insight into realistic algorithm performance*

“A Vida Como Ela É . . .”

O que fazer quando é preciso resolver instâncias concretas de problemas difíceis?

não existe manual com respostas prontas?



Skiena [2020]

DIMACS Implementation Challenges

> *The Challenges aid in determining realistic algorithm performance where worst-case analysis is overly pessimistic and probabilistic models are too unrealistic Experimentation can provide insight into realistic algorithm performance and it can provide new perspective that motivates deeper analytical results*

Programação Linear Inteira (Mista)

Programação Linear Inteira (Mista)

maior sucesso de todos?

Programação Linear Inteira (Mista)

maior sucesso de todos?

abertura de inteiras áreas de pesquisa em teoria

Programação Linear Inteira (Mista)

maior sucesso de todos?

abertura de inteiras áreas de pesquisa em teoria

impacto prático enorme

Programação Linear Inteira (Mista)

maior sucesso de todos?

abertura de inteiras áreas de pesquisa em teoria

impacto prático enorme

software efetivo e amplamente disponível

Resolvedores SAT

Resolvedores SAT

aplicações “industriais”, como PLI

Resolvedores SAT

aplicações “industriais”, como PLI

“matemática automatizada”

Resolvedores SAT

aplicações “industriais”, como PLI

“matemática automatizada”

- Conjectura das Triplas Pitagóricas Heule et al. [2016]

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático?

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

---

- "Schur Number Five" — Heule [2018]

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

---

- "Schur Number Five" — Heule [2018]
  $> n$ tal que $[n]$ pode ser 5-colorido sem $a = b + c$ monocromático?

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

---

- "Schur Number Five" — Heule [2018]
  $> n$ tal que $[n]$ pode ser 5-colorido sem $a = b + c$ monocromático? 160

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

---

- "Schur Number Five" — Heule [2018]
  $> n$ tal que $[n]$ pode ser 5-colorido sem $a = b + c$ monocromático? 160

- Conjectura de Keller em dimensão 7 — Brakensiek et al. [2020]

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

---

- "Schur Number Five" — Heule [2018]
  $> n$ tal que $[n]$ pode ser 5-colorido sem $a = b + c$ monocromático? 160

- Conjectura de Keller em dimensão 7 — Brakensiek et al. [2020]
  $\omega(G) < 128$ para todo $G \in \{G_1, G_2, G_3\}$?

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

---

- "Schur Number Five" — Heule [2018]
  $> n$ tal que $[n]$ pode ser 5-colorido sem $a = b + c$ monocromático? 160

- Conjectura de Keller em dimensão 7 — Brakensiek et al. [2020]
  $\omega(G) < 128$ para todo $G \in \{G_1, G_2, G_3\}$? não

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

---

- "Schur Number Five" — Heule [2018]
  $> n$ tal que $[n]$ pode ser 5-colorido sem $a = b + c$ monocromático? 160

- Conjectura de Keller em dimensão 7 — Brakensiek et al. [2020]
  $\omega(G) < 128$ para todo $G \in \{G_1, G_2, G_3\}$? não: conjectura é falsa

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

---

- "Schur Number Five" — Heule [2018]
  $> n$ tal que $[n]$ pode ser 5-colorido sem $a = b + c$ monocromático? 160

- Conjectura de Keller em dimensão 7 — Brakensiek et al. [2020]
  $\omega(G) < 128$ para todo $G \in \{G_1, G_2, G_3\}$? não: conjectura é falsa

---

- Conjectura de Collatz

Resolvedores SAT

aplicações "industriais", como PLI

"matemática automatizada"

- Conjectura das Triplas Pitagóricas — Heule et al. [2016]
  é possível bicolorir os naturais positivos sem $a^2 = b^2 + c^2$ monocromático? não

---

- "Schur Number Five" — Heule [2018]
  $> n$ tal que $[n]$ pode ser 5-colorido sem $a = b + c$ monocromático? 160

- Conjectura de Keller em dimensão 7 — Brakensiek et al. [2020]
  $\omega(G) < 128$ para todo $G \in \{G_1, G_2, G_3\}$? não: conjectura é falsa

---

- Conjectura de Collatz (tentativa) — Yolcu et al. [2021]

Programação por Restrições

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP

Gamrath et al. [2020]

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP — Gamrath et al. [2020]

Programação Linear

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP — Gamrath et al. [2020]

Programação Linear, Inteira

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP — Gamrath et al. [2020]

Programação Linear, Inteira, Mista

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP — Gamrath et al. [2020]

Programação Linear, Inteira, Mista, não-Linear

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP — Gamrath et al. [2020]

Programação Linear, Inteira, Mista, não-Linear, convexa

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP — Gamrath et al. [2020]

Programação Linear, Inteira, Mista, não-Linear, convexa, Programação por restrições

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP — Gamrath et al. [2020]

Programação Linear, Inteira, Mista, não-Linear, convexa, Programação por restrições...

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP — Gamrath et al. [2020]

Programação Linear, Inteira, Mista, não-Linear, convexa, Programação por restrições...

executável e biblioteca

Programação por Restrições

convergência das tecnologias de resolvedores PLI e SAT

SCIP — Gamrath et al. [2020]

Programação Linear, Inteira, Mista, não-Linear, convexa, Programação por restrições...

executável e biblioteca

"restrição" pode ser uma subrotina que responde se um ponto do espaço de busca é viável ou não

Outros (Slide de Utilidade Pública)

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]
  isomorfismo e grupos de automorfismos de grafos

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]
  isomorfismo e grupos de automorfismos de grafos
- Concorde

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]
  isomorfismo e grupos de automorfismos de grafos
- Concorde
  biblioteca para solução do Problema do Caixeiro Viajante simétrico

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]
  isomorfismo e grupos de automorfismos de grafos
- Concorde
  biblioteca para solução do Problema do Caixeiro Viajante simétrico serviço "online"

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]
  isomorfismo e grupos de automorfismos de grafos
- Concorde
  biblioteca para solução do Problema do Caixeiro Viajante simétrico serviço "online"
- calculadora "online" de decomposições arbóreas van Wersch and Kelk [2017]

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]
  isomorfismo e grupos de automorfismos de grafos
- Concorde
  biblioteca para solução do Problema do Caixeiro Viajante simétrico serviço "online"
- calculadora "online" de decomposições arbóreas van Wersch and Kelk [2017]
  programação dinâmica

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]
  isomorfismo e grupos de automorfismos de grafos
- Concorde
  biblioteca para solução do Problema do Caixeiro Viajante simétrico serviço "online"
- calculadora "online" de decomposições arbóreas van Wersch and Kelk [2017]
  programação dinâmica: guarda as decomposições já calculadas

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]
  isomorfismo e grupos de automorfismos de grafos
- Concorde
  biblioteca para solução do Problema do Caixeiro Viajante simétrico serviço "online"
- calculadora "online" de decomposições arbóreas van Wersch and Kelk [2017]
  programação dinâmica: guarda as decomposições já calculadas
- Karlsruhe High Quality Partitioning (KAHIP)

Outros (Slide de Utilidade Pública)

- Sagemath: [Stein and Joyner, 2005]
  integração diversos sistemas (otimização, computação simbólica etc) via Python
- NetworkX Hagberg et al. [2008]
  biblioteca Python para grafos: implementa vários algoritmos fundamentais
- nauty McKay and Piperno [2014]
  isomorfismo e grupos de automorfismos de grafos
- Concorde
  biblioteca para solução do Problema do Caixeiro Viajante simétrico serviço "online"
- calculadora "online" de decomposições arbóreas van Wersch and Kelk [2017]
  programação dinâmica: guarda as decomposições já calculadas
- Karlsruhe High Quality Partitioning (KAHIP)
  programas para problemas de particionamento de grafos

Eu tenho um sonho . . .

Eu tenho um sonho . . .

que um dia, possamos todos contar com implementações disponíveis

Eu tenho um sonho . . .

que um dia, possamos todos contar com implementações disponíveis para todos os problemas

Eu tenho um sonho . . .

que um dia, possamos todos contar com implementações disponíveis para todos os problemas independentemente de complexidade, cor da pele, gênero, crença,

Eu tenho um sonho . . .

que um dia, possamos todos contar com implementações disponíveis para todos os problemas independentemente de complexidade, cor da pele, gênero, crença, livres desde o código fonte

Problema da Clique Máxima: Motivação Inicial

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído: estudo de tolerância a falhas

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído: estudo de tolerância a falhas

simulação de redes de $200 \leq n \leq 461$ sistemas sujeitos a falhas

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído: estudo de tolerância a falhas

simulação de redes de $200 \leq n \leq 461$ sistemas sujeitos a falhas (PlanetLab)

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído: estudo de tolerância a falhas

simulação de redes de $200 \leq n \leq 461$ sistemas sujeitos a falhas (PlanetLab)

objetivo: avaliar o tamanho da "maior subrede sobrevivente"

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído: estudo de tolerância a falhas

simulação de redes de $200 \leq n \leq 461$ sistemas sujeitos a falhas (PlanetLab)

objetivo: avaliar o tamanho da "maior subrede sobrevivente" (= clique máxima)

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído: estudo de tolerância a falhas

simulação de redes de $200 \leq n \leq 461$ sistemas sujeitos a falhas (PlanetLab)

objetivo: avaliar o tamanho da "maior subrede sobrevivente" (= clique máxima)

grafo dinâmico de n vértices

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído: estudo de tolerância a falhas

simulação de redes de $200 \leq n \leq 461$ sistemas sujeitos a falhas (PlanetLab)

objetivo: avaliar o tamanho da "maior subrede sobrevivente" (= clique máxima)

grafo dinâmico de n vértices: a cada instante, arestas podem sumir ou aparecer

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído: estudo de tolerância a falhas

simulação de redes de $200 \leq n \leq 461$ sistemas sujeitos a falhas (PlanetLab)

objetivo: avaliar o tamanho da "maior subrede sobrevivente" (= clique máxima)

grafo dinâmico de n vértices: a cada instante, arestas podem sumir ou aparecer

não é grafo aleatório

Problema da Clique Máxima: Motivação Inicial

simulação de sistema distribuído: estudo de tolerância a falhas

simulação de redes de $200 \leq n \leq 461$ sistemas sujeitos a falhas (PlanetLab)

objetivo: avaliar o tamanho da "maior subrede sobrevivente" (= clique máxima)

grafo dinâmico de n vértices: a cada instante, arestas podem sumir ou aparecer

não é grafo aleatório

objetivo: determinar tamanho da maior clique a cada instante da simulação

Trocando em miúdos

Trocando em miúdos

encontrar clique máxima em 8448 grafos

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`: implementação "de calibragem" DIMACS Implementation Challenge 1992-1993

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`: implementação "de calibragem" DIMACS Implementation Challenge 1992-1993

> *"a simple-minded branch-and-bound algorithm very similar to that of Carraghan and Pardalos [1990]"*

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`: implementação "de calibragem" DIMACS Implementation Challenge 1992-1993

> *"a simple-minded branch-and-bound algorithm very similar to that of Carraghan and Pardalos [1990]"*

versão final

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`: implementação "de calibragem" DIMACS Implementation Challenge 1992-1993

> *"a simple-minded branch-and-bound algorithm very similar to that of Carraghan and Pardalos [1990]"*

versão final: algoritmo de Tomita and Kameda [2007]

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`: implementação "de calibragem" DIMACS Implementation Challenge 1992-1993

> *"a simple-minded branch-and-bound algorithm very similar to that of Carraghan and Pardalos [1990]"*

versão final: algoritmo de Tomita and Kameda [2007] C++ usando Boost Graph Library

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`: implementação "de calibragem" DIMACS Implementation Challenge 1992-1993

> *"a simple-minded branch-and-bound algorithm very similar to that of Carraghan and Pardalos [1990]"*

versão final: algoritmo de Tomita and Kameda [2007] C++ usando Boost Graph Library

cliques máximas entre 79 e 257 vértices

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`: implementação "de calibragem" DIMACS Implementation Challenge 1992-1993

> *"a simple-minded branch-and-bound algorithm very similar to that of Carraghan and Pardalos [1990]"*

versão final: algoritmo de Tomita and Kameda [2007] C++ usando Boost Graph Library

cliques máximas entre 79 e 257 vértices

história completa: Duarte Jr. et al. [2010]

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`: implementação "de calibragem" DIMACS Implementation Challenge 1992-1993

> *"a simple-minded branch-and-bound algorithm very similar to that of Carraghan and Pardalos [1990]"*

versão final: algoritmo de Tomita and Kameda [2007] C++ usando Boost Graph Library

cliques máximas entre 79 e 257 vértices

história completa: Duarte Jr. et al. [2010]

menos de 1 segundo para resolver cada instância

Trocando em miúdos

encontrar clique máxima em 8448 grafos entre 200 e 461 vértices

avaliação preliminar: `dfmax`: implementação "de calibragem" DIMACS Implementation Challenge 1992-1993

> *"a simple-minded branch-and-bound algorithm very similar to that of Carraghan and Pardalos [1990]"*

versão final: algoritmo de Tomita and Kameda [2007] C++ usando Boost Graph Library

cliques máximas entre 79 e 257 vértices

história completa: Duarte Jr. et al. [2010]

menos de 1 segundo para resolver cada instância 😮

Vamos combinar . . .

Vamos combinar . . .

G: grafo de n vértices e m arestas

Vamos combinar . . .

G: grafo de n vértices e m arestas

clique: conjunto de vértices que induz grafo completo

Vamos combinar . . .

G: grafo de n vértices e m arestas

clique: conjunto de vértices que induz grafo completo
não necessariamente maximal

Vamos combinar . . .

G: grafo de n vértices e m arestas

clique: conjunto de vértices que induz grafo completo
não necessariamente maximal

$\omega(G)$: tamanho máximo de uma clique em G

Vamos combinar . . .

G: grafo de n vértices e m arestas

clique: conjunto de vértices que induz grafo completo
não necessariamente maximal

$\omega(G)$: tamanho máximo de uma clique em G

Problema da Clique: dados G, k, responder se "G tem clique de tamanho k?"

Vamos combinar . . .

G: grafo de n vértices e m arestas

clique: conjunto de vértices que induz grafo completo
não necessariamente maximal

$\omega(G)$: tamanho máximo de uma clique em G

Problema da Clique: dados G, k, responder se "G tem clique de tamanho k?"
(CLIQUE)

Vamos combinar . . .

G: grafo de n vértices e m arestas

clique: conjunto de vértices que induz grafo completo
não necessariamente maximal

$\omega(G)$: tamanho máximo de uma clique em G

Problema da Clique: dados G, k, responder se "G tem clique de tamanho k?"
(CLIQUE)

Problema da Clique Máxima: determinar uma clique de tamanho máximo em um grafo dado

Vamos combinar . . .

G: grafo de n vértices e m arestas

clique: conjunto de vértices que induz grafo completo
não necessariamente maximal

$\omega(G)$: tamanho máximo de uma clique em G

Problema da Clique: dados G, k, responder se "G tem clique de tamanho k?"
(CLIQUE)

Problema da Clique Máxima: determinar uma clique de tamanho máximo em um grafo dado (MaxClique)

Vamos combinar . . .

G: grafo de n vértices e m arestas

clique: conjunto de vértices que induz grafo completo
não necessariamente maximal

$\omega(G)$: tamanho máximo de uma clique em G

Problema da Clique: dados G, k, responder se "G tem clique de tamanho k?"
(CLIQUE)

Problema da Clique Máxima: determinar uma clique de tamanho máximo em um grafo dado (MaxClique)

lg $:= \log_2$

Complexidade

Complexidade

CLIQUE é $\mathcal{NP}$-completo Karp [1972]

Complexidade

CLIQUE é $\mathcal{NP}$-completo Karp [1972]

MaxClique é $\mathcal{NP}$-difícil

Complexidade

CLIQUE é $\mathcal{NP}$-completo — Karp [1972]

MaxClique é $\mathcal{NP}$-difícil

MaxClique é inaproximável a menos de $n^{1-\varepsilon}$, para todo $\varepsilon > 0$ — Zuckerman [2006]

Complexidade

CLIQUE é $\mathcal{NP}$-completo — Karp [1972]

MaxClique é $\mathcal{NP}$-difícil

MaxClique é inaproximável a menos de $n^{1-\varepsilon}$, para todo $\varepsilon > 0$ — Zuckerman [2006]

CLIQUE é $W[1]$–completo — Downey and Fellows [1995]

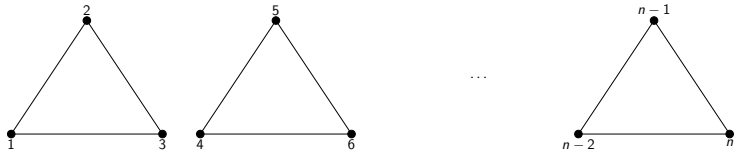
“When in doubt, use brute force” (Ken Thompson)

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

“When in doubt, use brute force” (Ken Thompson)

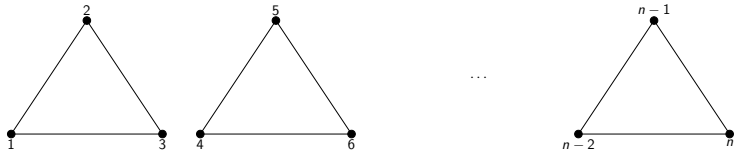
grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]



- G: $\frac{n}{3}$ triângulos

“When in doubt, use brute force” (Ken Thompson)

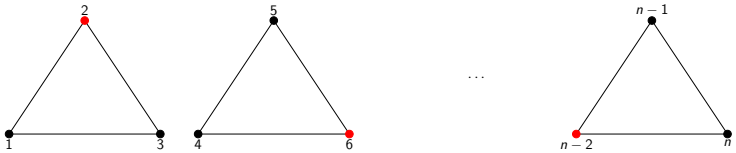
grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]



- G: $\frac{n}{3}$ triângulos
- conjunto independente maximal:

“When in doubt, use brute force” (Ken Thompson)

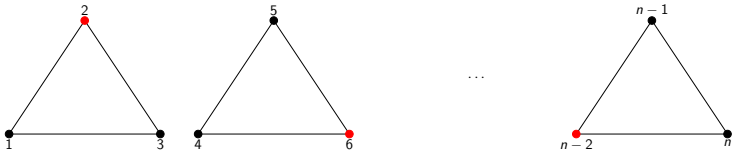
grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]



- G: $\frac{n}{3}$ triângulos
- conjunto independente maximal: 1 vértice em cada triângulo

“When in doubt, use brute force” (Ken Thompson)

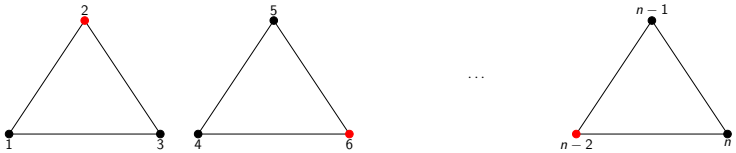
grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]



- G: $\frac{n}{3}$ triângulos
- conjunto independente maximal: 1 vértice em cada triângulo
- $3^{n/3}$ conjuntos independentes maximais

“When in doubt, use brute force” (Ken Thompson)

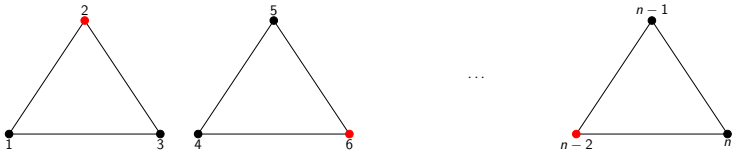
grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]



- G: $\frac{n}{3}$ triângulos
- conjunto independente maximal: 1 vértice em cada triângulo
- $3^{n/3}$ conjuntos independentes maximais
- $\overline{G}$:

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]



- G: $\frac{n}{3}$ triângulos
- conjunto independente maximal: 1 vértice em cada triângulo
- $3^{n/3}$ conjuntos independentes maximais
- $\overline{G}$: $3^{n/3}$ cliques maximais

“When in doubt, use brute force” (Ken Thompson)

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

força bruta

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

força bruta $\rightarrow$ enumerar cliques maximais

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

força bruta $\rightarrow$ enumerar cliques maximais $\Omega(3^{n/3})$ no pior caso

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

força bruta $\rightarrow$ enumerar cliques maximais $\Omega(3^{n/3})$ no pior caso

enumeração de todas as cliques maximais

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

força bruta $\rightarrow$ enumerar cliques maximais $\Omega(3^{n/3})$ no pior caso

enumeração de todas as cliques maximais

- primeiro algoritmo: Bron and Kerbosch [1973]

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

força bruta $\rightarrow$ enumerar cliques maximais $\Omega(3^{n/3})$ no pior caso

enumeração de todas as cliques maximais

- primeiro algoritmo: Bron and Kerbosch [1973]
- em tempo $O(3^{n/3})$: Tomita et al. [2006]

"When in doubt, use brute force" (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

força bruta $\rightarrow$ enumerar cliques maximais $\Omega(3^{n/3})$ no pior caso

enumeração de todas as cliques maximais

- primeiro algoritmo: Bron and Kerbosch [1973]
- em tempo $O(3^{n/3})$: Tomita et al. [2006], melhorado por Naudé [2015]

“When in doubt, use brute force” (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

força bruta $\rightarrow$ enumerar cliques maximais $\Omega(3^{n/3})$ no pior caso

enumeração de todas as cliques maximais

- primeiro algoritmo: Bron and Kerbosch [1973]
- em tempo $O(3^{n/3})$: Tomita et al. [2006], melhorado por Naudé [2015]
- em tempo $O(m(\#\text{ cliques maximais}))$: Hou et al. [2016]

"When in doubt, use brute force" (Ken Thompson)

grafo de n vértices pode ter $3^{n/3}$ cliques maximais distintas Moon and Moser [1965]

força bruta $\rightarrow$ enumerar cliques maximais $\Omega(3^{n/3})$ no pior caso

enumeração de todas as cliques maximais

- primeiro algoritmo: Bron and Kerbosch [1973]
- em tempo $O(3^{n/3})$: Tomita et al. [2006], melhorado por Naudé [2015]
- em tempo $O(m(\#\text{ cliques maximais}))$: Hou et al. [2016]

> *"These days, though, you have to be pretty technical before you can even aspire to crudeness."* *Gibson [1995]*

MaxClique em tempo $o(3^{n/3})$

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976]

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976]

$2^{n/3} = (1.25\ldots)^n$

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976]

$2^{n/3} = (1.25\ldots)^n$

$O(2^{n/4})$ Robson [2001]

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976] $2^{n/3} = (1.25\ldots)^n$

$O(2^{n/4})$ Robson [2001] espaço exponencial $2^{n/4} = (1.18\ldots)^n$

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976] $2^{n/3} = (1.25\ldots)^n$

$O(2^{n/4})$ Robson [2001] espaço exponencial $2^{n/4} = (1.18\ldots)^n$

$O(2^{0.288n})$: Fomin et al. [2006]

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976] $2^{n/3} = (1.25\ldots)^n$

$O(2^{n/4})$ Robson [2001] espaço exponencial $2^{n/4} = (1.18\ldots)^n$

$O(2^{0.288n})$: Fomin et al. [2006] $2^{0.288n} = (1.22\ldots)^n$

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976] $2^{n/3} = (1.25\ldots)^n$

$O(2^{n/4})$ Robson [2001] espaço exponencial $2^{n/4} = (1.18\ldots)^n$

$O(2^{0.288n})$: Fomin et al. [2006] $2^{0.288n} = (1.22\ldots)^n$

$O^*(1.1966^n)$: Xiao and Nagamochi [2017]

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976] $2^{n/3} = (1.25\ldots)^n$

$O(2^{n/4})$ Robson [2001] espaço exponencial $2^{n/4} = (1.18\ldots)^n$

$O(2^{0.288n})$: Fomin et al. [2006] $2^{0.288n} = (1.22\ldots)^n$

$O^*(1.1966^n)$: Xiao and Nagamochi [2017] $2^{0.258n} < 1.1966^n < 2^{0.259n}$

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976]	$2^{n/3} = (1.25\ldots)^n$
$O(2^{n/4})$ Robson [2001] espaço exponencial	$2^{n/4} = (1.18\ldots)^n$
$O(2^{0.288n})$: Fomin et al. [2006]	$2^{0.288n} = (1.22\ldots)^n$
$O^*(1.1966^n)$: Xiao and Nagamochi [2017]	$2^{0.258n} < 1.1966^n < 2^{0.259n}$

- branch and bound

MaxClique em tempo $o(3^{n/3})$

$O(2^{n/3})$: Tarjan and Trojanowski [1976]	$2^{n/3} = (1.25\ldots)^n$
$O(2^{n/4})$ Robson [2001] espaço exponencial	$2^{n/4} = (1.18\ldots)^n$
$O(2^{0.288n})$: Fomin et al. [2006]	$2^{0.288n} = (1.22\ldots)^n$
$O^*(1.1966^n)$: Xiao and Nagamochi [2017]	$2^{0.258n} < 1.1966^n < 2^{0.259n}$

- branch and bound
- sem implementação

MaxClique em tempo $O(3^{n/3})$

MaxClique em tempo $O(3^{n/3})$

- Carraghan and Pardalos [1990]
- Fahle [2002], Östergård [2002]
- Tomita and Seki [2003]
- Konc and Janežič [2007], Tomita and Kameda [2007]
- Tomita et al. [2010], Li and Quan [2010a,b]
- San Segundo et al. [2011],
- Li et al. [2013], Maslov et al. [2013]
- San Segundo and Tapia [2014],
- San Segundo et al. [2016a,b], Tomita et al. [2016]
- San Segundo et al. [2017]

MaxClique em tempo $O(3^{n/3})$

- Carraghan and Pardalos [1990]
- Fahle [2002], Östergård [2002]
- Tomita and Seki [2003]
- Konc and Janežič [2007], Tomita and Kameda [2007]
- Tomita et al. [2010], Li and Quan [2010a,b]
- San Segundo et al. [2011],
- Li et al. [2013], Maslov et al. [2013]
- San Segundo and Tapia [2014],
- San Segundo et al. [2016a,b], Tomita et al. [2016]
- San Segundo et al. [2017]

- branch and bound

MaxClique em tempo $O(3^{n/3})$

- Carraghan and Pardalos [1990]
- Fahle [2002], Östergård [2002]
- Tomita and Seki [2003]
- Konc and Janežič [2007], Tomita and Kameda [2007]
- Tomita et al. [2010], Li and Quan [2010a,b]
- San Segundo et al. [2011],
- Li et al. [2013], Maslov et al. [2013]
- San Segundo and Tapia [2014],
- San Segundo et al. [2016a,b], Tomita et al. [2016]
- San Segundo et al. [2017]

- branch and bound
- sem análise

Metodologia

Metodologia

"benchmarking"

- tempo de execução

Metodologia

"benchmarking"

- tempo de execução
- tamanho da árvore de busca

Metodologia

"benchmarking"

- tempo de execução
- tamanho da árvore de busca (alguns)

Metodologia

"benchmarking"

- tempo de execução
- tamanho da árvore de busca (alguns)

tempo de execução medido em ambientes computacionais diferentes

Metodologia

"benchmarking"

- tempo de execução
- tamanho da árvore de busca (alguns)

tempo de execução medido em ambientes computacionais diferentes

fator de correção

Metodologia

"benchmarking"

- tempo de execução
- tamanho da árvore de busca (alguns)

tempo de execução medido em ambientes computacionais diferentes

fator de correção

1. executa `dfmax` num conjunto padrão de instâncias

Metodologia

"benchmarking"

- tempo de execução
- tamanho da árvore de busca (alguns)

tempo de execução medido em ambientes computacionais diferentes

fator de correção

1. executa `dfmax` num conjunto padrão de instâncias
2. usa tempos de execução para determinar constante para "normalizar" tempos

Metodologia

"benchmarking"

- tempo de execução
- tamanho da árvore de busca (alguns)

tempo de execução medido em ambientes computacionais diferentes

fator de correção

1. executa `dfmax` num conjunto padrão de instâncias
2. usa tempos de execução para determinar constante para "normalizar" tempos

implícito

Metodologia

"benchmarking"

- tempo de execução
- tamanho da árvore de busca (alguns)

tempo de execução medido em ambientes computacionais diferentes

fator de correção

1. executa `dfmax` num conjunto padrão de instâncias
2. usa tempos de execução para determinar constante para "normalizar" tempos

implícito: diferença de tempos de execução entre ambientes computacionais é constante multiplicativa

Metodologia

"benchmarking"

- tempo de execução
- tamanho da árvore de busca (alguns)

tempo de execução medido em ambientes computacionais diferentes

fator de correção

1. executa `dfmax` num conjunto padrão de instâncias
2. usa tempos de execução para determinar constante para "normalizar" tempos

implícito: diferença de tempos de execução entre ambientes computacionais é constante multiplicativa

não funciona: não é recomendável Prosser [2012]

Instâncias

Instâncias

grafos aleatórios

Instâncias

grafos aleatórios: $G_{n,p}$ com p constante

Instâncias

grafos aleatórios: $G_{n,p}$ com p constante

DIMACS Implementation Challenge 1992-1993

Instâncias

grafos aleatórios: $G_{n,p}$ com p constante

DIMACS Implementation Challenge 1992-1993: 66 grafos

Instâncias

grafos aleatórios: $G_{n,p}$ com p constante

DIMACS Implementation Challenge 1992-1993: 66 grafos (9 famílias)

Instâncias

grafos aleatórios: $G_{n,p}$ com p constante

DIMACS Implementation Challenge 1992-1993: 66 grafos (9 famílias) contribuições para o desafio

Instâncias

grafos aleatórios: $G_{n,p}$ com p constante

DIMACS Implementation Challenge 1992-1993: 66 grafos (9 famílias) contribuições para o desafio

BHOSLIB

Instâncias

grafos aleatórios: $G_{n,p}$ com p constante

DIMACS Implementation Challenge 1992-1993: 66 grafos (9 famílias) contribuições para o desafio

BHOSLIB: Benchmarks with Hidden Optimum Solutions for Graph Problems

Panorama

Panorama

resultados teóricos desanimadores

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

- nada

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

- nada: não há nada de surpreendente

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

- nada: não há nada de surpreendente
- instâncias

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

- nada: não há nada de surpreendente
- instâncias são “fáceis”

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

- nada: não há nada de surpreendente
- instâncias são “fáceis” (como são as “difíceis”)?

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

- nada: não há nada de surpreendente
- instâncias são "fáceis" (como são as "difíceis")?
- como interpretar as comparações entre os algoritmos/implementações?

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

- nada: não há nada de surpreendente
- instâncias são "fáceis" (como são as "difíceis")?
- como interpretar as comparações entre os algoritmos/implementações? há diferença significativa?

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

- nada: não há nada de surpreendente
- instâncias são "fáceis" (como são as "difíceis")?
- como interpretar as comparações entre os algoritmos/implementações? há diferença significativa? ou são flutuações fortuitas?

Panorama

resultados teóricos desanimadores

resultados experimentais animadores

o que acontece?

- nada: não há nada de surpreendente
- instâncias são "fáceis" (como são as "difíceis")?
- como interpretar as comparações entre os algoritmos/implementações?
  há diferença significativa? ou são flutuações fortuitas?
- como se relacionam os algoritmos implementados com os algoritmos analisados?

Branch & Bound para MaxClique: esquema

Branch & Bound para MaxClique: esquema

clique máxima em G

Branch & Bound para MaxClique: esquema

clique máxima em G: maior dentre

Branch & Bound para MaxClique: esquema

clique máxima em G: maior dentre

- clique máxima com v

Branch & Bound para MaxClique: esquema

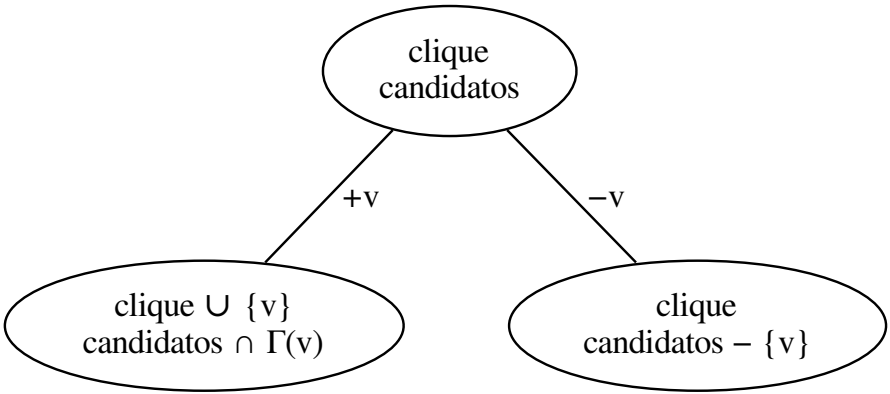
clique máxima em G: maior dentre

- clique máxima com v
- clique máxima sem v

Branch & Bound para MaxClique: esquema

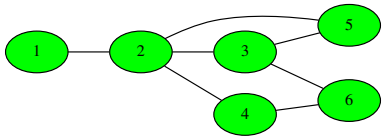
clique máxima em G: maior dentre

- clique máxima com v
- clique máxima sem v



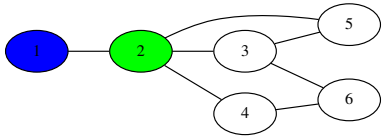
Branch & Bound para MaxClique: exemplo “sem bound”

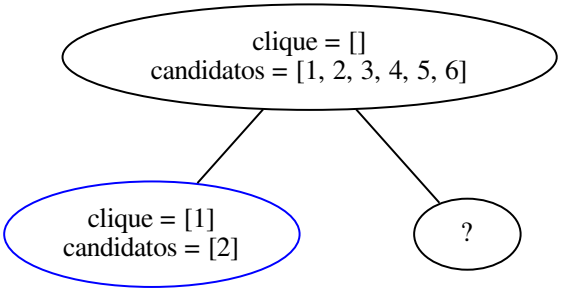
Branch & Bound para MaxClique: exemplo “sem bound”



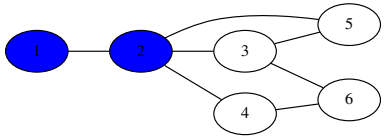
clique = []
candidatos = [1, 2, 3, 4, 5, 6]

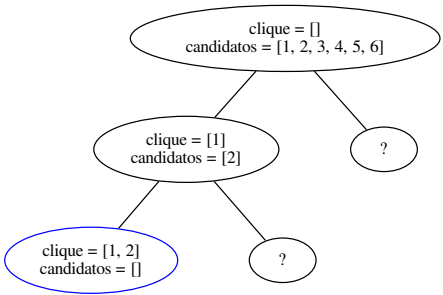
Branch & Bound para MaxClique: exemplo “sem bound”



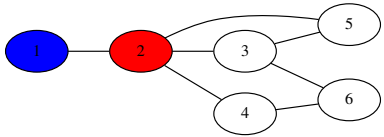


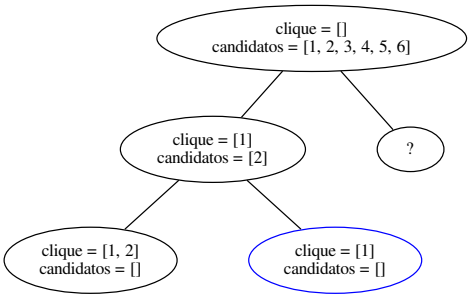
Branch & Bound para MaxClique: exemplo “sem bound”



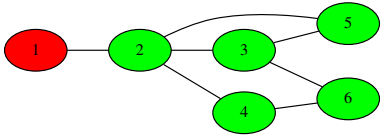


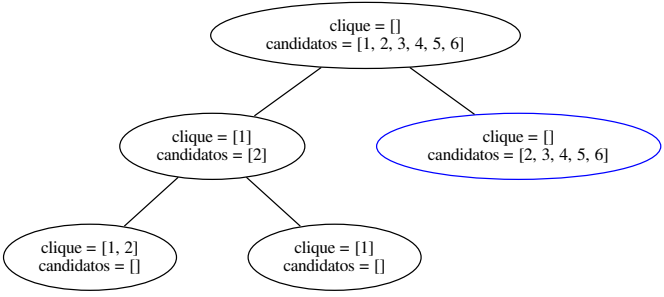
Branch & Bound para MaxClique: exemplo “sem bound”



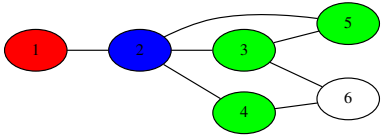


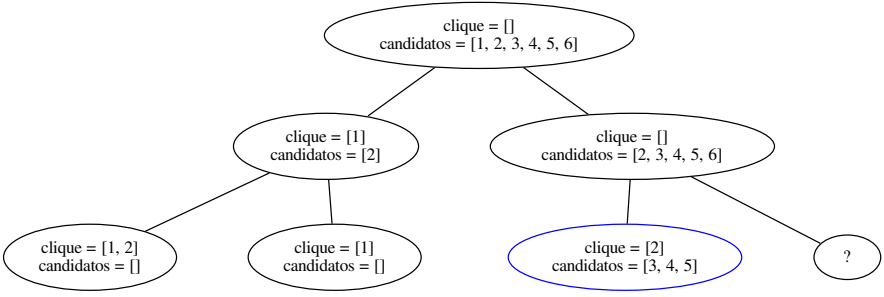
Branch & Bound para MaxClique: exemplo “sem bound”



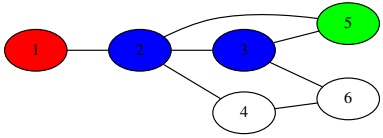


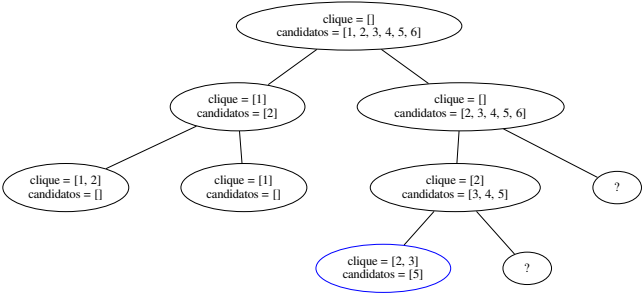
Branch & Bound para MaxClique: exemplo “sem bound”



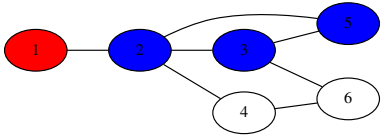


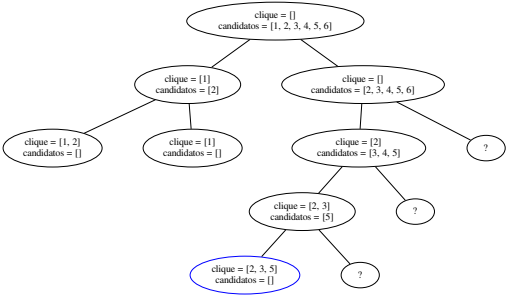
Branch & Bound para MaxClique: exemplo “sem bound”



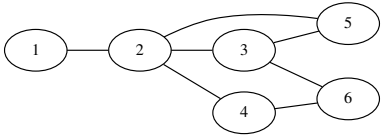


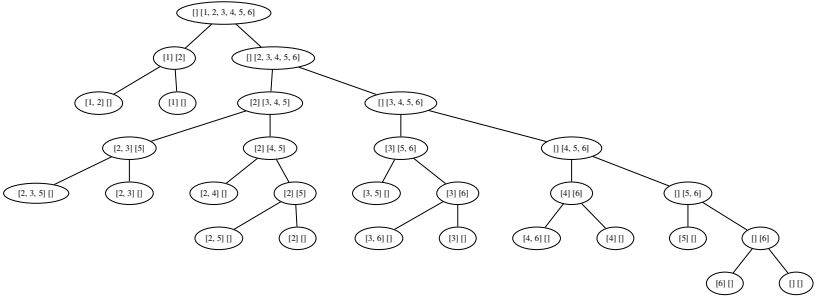
Branch & Bound para MaxClique: exemplo "sem bound"



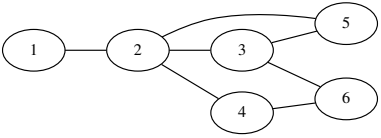


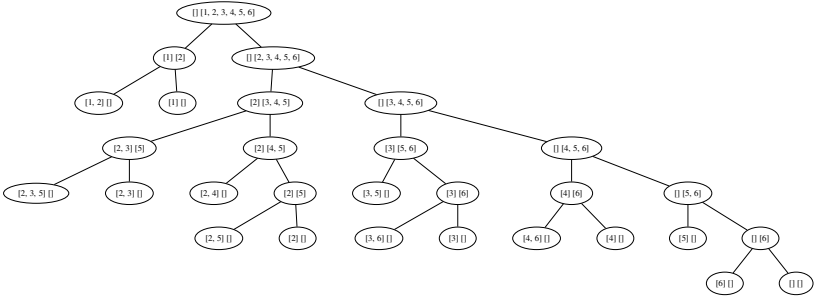
Branch & Bound para MaxClique: exemplo "sem bound"





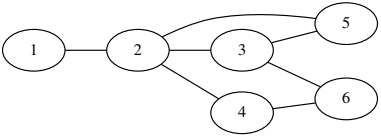
Branch & Bound para MaxClique: exemplo "sem bound"

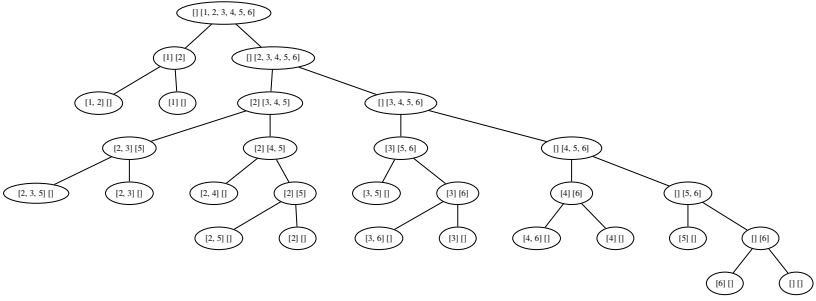




Algoritmo nobound

Branch & Bound para MaxClique: exemplo "sem bound"

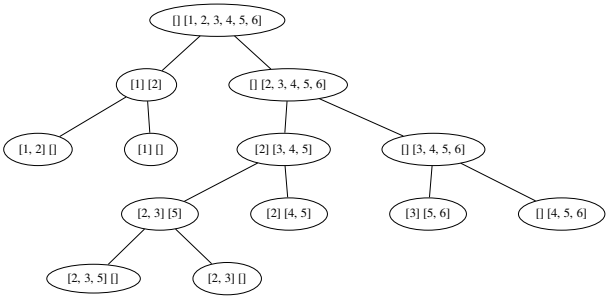




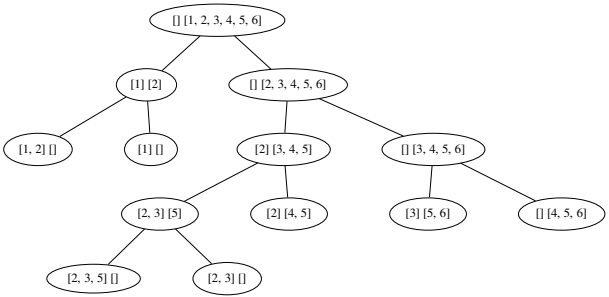
Algoritmo nobound: $|T_{\text{nobound}}(G)| = 29$

Branch & Bound para MaxClique: bound = # candidatos

Branch & Bound para MaxClique: bound = # candidatos

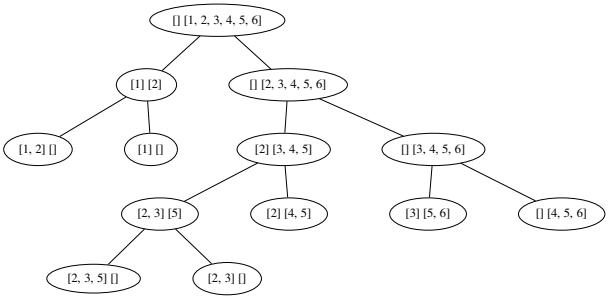


Branch & Bound para MaxClique: bound = # candidatos



Algoritmo basic

Branch & Bound para MaxClique: bound = # candidatos



Algoritmo basic: $|T_{\text{basic}}(G)| = 13$

Branch & Bound para MaxClique

Branch & Bound para MaxClique

MaxClique(G, Q, K, C)

Se $K = \emptyset$
    Se $|Q| > |C|$
        $C \leftarrow Q$
Senão
    Se $|Q| + \textbf{bound}(G, K) > |C|$
    $v \leftarrow$ elemento de K (pivô)
    $C \leftarrow$ MaxClique$(G, Q \cup \{v\}, K \cap N(v), C)$
    $C \leftarrow$ MaxClique$(G, Q, K - \{v\}, C)$
Devolva C

Diferentes Algoritmos

Diferentes Algoritmos

principais algoritmos tem estrutura parecida

Diferentes Algoritmos

principais algoritmos tem estrutura parecida

podem ser expressos como “instâncias” de um “algoritmo parametrizado”

Diferentes Algoritmos

principais algoritmos tem estrutura parecida

podem ser expressos como “instâncias” de um “algoritmo parametrizado”

- escolha do pivô

Diferentes Algoritmos

principais algoritmos tem estrutura parecida

podem ser expressos como “instâncias” de um “algoritmo parametrizado”

- escolha do pivô
- função que computa o “bound”

Diferentes Algoritmos

principais algoritmos tem estrutura parecida

podem ser expressos como “instâncias” de um “algoritmo parametrizado”

- escolha do pivô
- função que computa o “bound”: coloração

Diferentes Algoritmos

principais algoritmos tem estrutura parecida

podem ser expressos como “instâncias” de um “algoritmo parametrizado”

- escolha do pivô
- função que computa o “bound”: coloração
- pré e pós-processamento dos conjuntos

Diferentes Algoritmos

principais algoritmos tem estrutura parecida

podem ser expressos como “instâncias” de um “algoritmo parametrizado”

- escolha do pivô
- função que computa o “bound”: coloração
- pré e pós-processamento dos conjuntos
- escolha do próximo nó da árvore a visitar

Diferentes Algoritmos

principais algoritmos tem estrutura parecida

podem ser expressos como “instâncias” de um “algoritmo parametrizado”

- escolha do pivô
- função que computa o “bound”: coloração
- pré e pós-processamento dos conjuntos
- escolha do próximo nó da árvore a visitar (subproblema a examinar)

MCBB

MCBB

família de algoritmos que “se encaixam” nesse esquema

MCBB

família de algoritmos que “se encaixam” nesse esquema

implementado em Python Züge [2011], Carmo and Züge [2012]

MCBB

família de algoritmos que “se encaixam” nesse esquema

implementado em Python Züge [2011], Carmo and Züge [2012]

e C++ Correa [2020], Carmo et al. [2020]

MCBB

família de algoritmos que “se encaixam” nesse esquema

implementado em Python Züge [2011], Carmo and Züge [2012]

e C++ Correa [2020], Carmo et al. [2020]

https://gitlab.c3sl.ufpr.br/apzuge/maxcliquebb

MCBB

família de algoritmos que “se encaixam” nesse esquema

implementado em Python Züge [2011], Carmo and Züge [2012]

e C++ Correa [2020], Carmo et al. [2020]

https://gitlab.c3sl.ufpr.br/apzuge/maxcliquebb

não procura eficiência

MCBB

família de algoritmos que “se encaixam” nesse esquema

implementado em Python Züge [2011], Carmo and Züge [2012]

e C++ Correa [2020], Carmo et al. [2020]

https://gitlab.c3sl.ufpr.br/apzuge/maxcliquebb

não procura eficiência: facilidade de análise experimental

MCBB

família de algoritmos que “se encaixam” nesse esquema

implementado em Python Züge [2011], Carmo and Züge [2012]

e C++ Correa [2020], Carmo et al. [2020]

https://gitlab.c3sl.ufpr.br/apzuge/maxcliquebb

não procura eficiência: facilidade de análise experimental

permite comparação no mesmo ambiente computacional (SO, linguagem, hardware)

MCBB

família de algoritmos que “se encaixam” nesse esquema

implementado em Python Züge [2011], Carmo and Züge [2012]

e C++ Correa [2020], Carmo et al. [2020]

https://gitlab.c3sl.ufpr.br/apzuge/maxcliquebb

não procura eficiência: facilidade de análise experimental

permite comparação no mesmo ambiente computacional (SO, linguagem, hardware)

primeiro passo em direção a uma análise experimental mais estruturada

Análise Experimental de Algoritmos

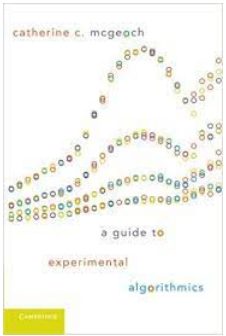
David Johnson

Análise Experimental de Algoritmos

David Johnson
(“patrono”)

Análise Experimental de Algoritmos

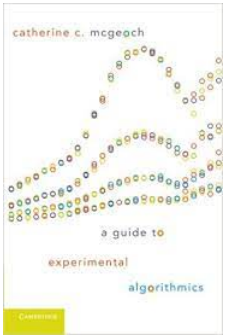
David Johnson
("patrono")



McGeoch [2012]

Análise Experimental de Algoritmos

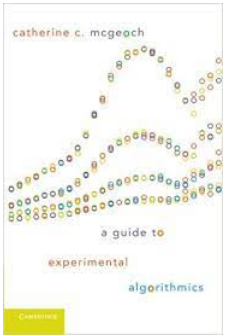
David Johnson
("patrono")



McGeoch [2012]
reúne/estrutura conhecimento até então disperso em artigos

Análise Experimental de Algoritmos

David Johnson
("patrono")



McGeoch [2012]
reúne/estrutura conhecimento até então disperso em artigos

dos Anjos [2015], dos Anjos et al. [2016]: explica e aplica ideias a MaxClique

Instâncias Difíceis

Instâncias Difíceis

Lavnikevich [2013]: família de instâncias para as quais algoritmos usando coloração como único "bound" tomam tempo $\Omega(2^{n/5})$

Instâncias Difíceis

Lavnikevich [2013]: família de instâncias para as quais algoritmos usando coloração como único "bound" tomam tempo $\Omega(2^{n/5})$ $2^{n/5} = (1.14\ldots)^n$

Instâncias Difíceis

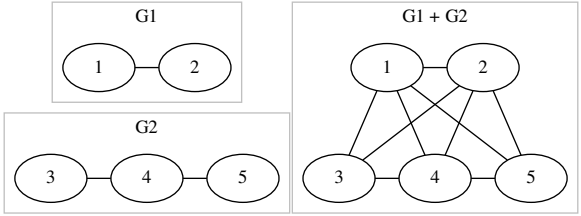
Lavnikevich [2013]: família de instâncias para as quais algoritmos usando coloração como único "bound" tomam tempo $\Omega(2^{n/5})$ $2^{n/5} = (1.14\ldots)^n$

$G_q =$ "join" de q cópias de C_5

Instâncias Difíceis

Lavnikevich [2013]: família de instâncias para as quais algoritmos usando coloração como único "bound" tomam tempo $\Omega(2^{n/5})$ $\qquad 2^{n/5} = (1.14\ldots)^n$

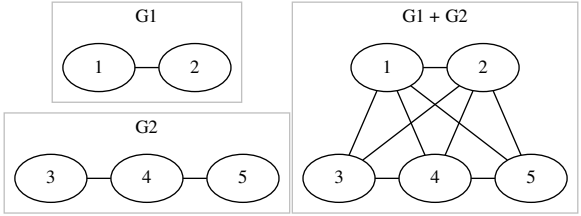
$G_q =$ "join" de q cópias de C_5



Instâncias Difíceis

Lavnikevich [2013]: família de instâncias para as quais algoritmos usando coloração como único "bound" tomam tempo $\Omega(2^{n/5})$ $\qquad 2^{n/5} = (1.14\ldots)^n$

$G_q =$ "join" de q cópias de C_5

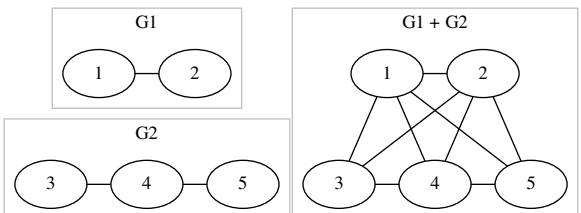


$\chi(G_q) - \omega(G_q) = q$

Instâncias Difíceis

Lavnikevich [2013]: família de instâncias para as quais algoritmos usando coloração como único "bound" tomam tempo $\Omega(2^{n/5})$ $\qquad 2^{n/5} = (1.14\ldots)^n$

$G_q =$ "join" de q cópias de C_5



$\chi(G_q) - \omega(G_q) = q$

reconhecíveis/contornáveis em tempo polinomial Züge and Carmo [2016a]

Cliques em Grafos Aleatórios

Cliques em Grafos Aleatórios

grafo aleatório

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$: espaço uniforme

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$: espaço uniforme: todo grafo de n vértices tem a mesma probabilidade

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$: espaço uniforme: todo grafo de n vértices tem a mesma probabilidade

$\omega(G)$: "variável quase determinada"

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$: espaço uniforme: todo grafo de n vértices tem a mesma probabilidade

$\omega(G)$: "variável quase determinada"

para quase todo G

$$\omega(G) = 2 \lg n$$

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$: espaço uniforme: todo grafo de n vértices tem a mesma probabilidade

$\omega(G)$: "variável quase determinada"

para quase todo G

$$\omega(G) = 2 \lg n + O(\log \log n)$$

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$: espaço uniforme: todo grafo de n vértices tem a mesma probabilidade

$\omega(G)$: "variável quase determinada"

para quase todo G

$$\omega(G) = 2 \lg n + O(\log \log n) = 2 \lg n - 2 \lg lgn + 1.89 \ldots + o(1)$$

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$: espaço uniforme: todo grafo de n vértices tem a mesma probabilidade

$\omega(G)$: "variável quase determinada"

para quase todo G

$$\omega(G) = 2 \lg n + O(\log \log n) = 2 \lg n - 2 \lg lgn + 1.89 \ldots + o(1)$$

$\mathbb{P}\,(\omega(G)$ diferente disso)

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$: espaço uniforme: todo grafo de n vértices tem a mesma probabilidade

$\omega(G)$: "variável quase determinada"

para quase todo G

$$\omega(G) = 2 \lg n + O(\log \log n) = 2 \lg n - 2 \lg lgn + 1.89 \ldots + o(1)$$

$\mathbb{P}\,(\omega(G)$ diferente disso) cai **muito rapidamente** com a diferença

Cliques em Grafos Aleatórios

grafo aleatório: $\mathcal{G}_{n,p}$, p constante

$p = 1/2$: espaço uniforme: todo grafo de n vértices tem a mesma probabilidade

$\omega(G)$: "variável quase determinada"

para quase todo G

$$\omega(G) = 2 \lg n + O(\log \log n) = 2 \lg n - 2 \lg lgn + 1.89 \ldots + o(1)$$

$\mathbb{P}\,(\omega(G)$ diferente disso) cai **muito rapidamente** com a diferença

detalhes: [Bollobás, 2001, Cap. 11]

Cliques em Grafos Aleatórios

Cliques em Grafos Aleatórios

resultados assintóticos

Cliques em Grafos Aleatórios

resultados assintóticos com efeitos “visíveis a olho nu”

Cliques em Grafos Aleatórios

resultados assintóticos com efeitos “visíveis a olho nu”

- clique máxima tem tamanho logarítmico

Cliques em Grafos Aleatórios

resultados assintóticos com efeitos "visíveis a olho nu"

- clique máxima tem tamanho logarítmico
- as cliques maximais tem tamanho entre $\lg n$ e $2 \lg n$

Cliques em Grafos Aleatórios

resultados assintóticos com efeitos "visíveis a olho nu"

- clique máxima tem tamanho logarítmico
- as cliques maximais tem tamanho entre $\lg n$ e $2 \lg n$

panorama homogêneo

Cliques em Grafos Aleatórios

resultados assintóticos com efeitos "visíveis a olho nu"

- clique máxima tem tamanho logarítmico
- as cliques maximais tem tamanho entre $\lg n$ e $2 \lg n$

panorama homogêneo $\times$ "testar o algoritmo com grafos aleatórios"

Cliques em Grafos Aleatórios

resultados assintóticos com efeitos "visíveis a olho nu"

- clique máxima tem tamanho logarítmico
- as cliques maximais tem tamanho entre $\lg n$ e $2 \lg n$

panorama homogêneo $\times$ "testar o algoritmo com grafos aleatórios"

"distorce" (a interpretação de) resultados experimentais?

Programação Dinâmica

Programação Dinâmica

Chvátal [1977]

Programação Dinâmica

Chvátal [1977]: família de algoritmos de programação dinâmica

Programação Dinâmica

Chvátal [1977]: família de algoritmos de programação dinâmica para Conjunto Independente Máximo

Programação Dinâmica

Chvátal [1977]: família de algoritmos de programação dinâmica para Conjunto Independente Máximo

executam em tempo subexponencial para quase todo grafo

Programação Dinâmica

Chvátal [1977]: família de algoritmos de programação dinâmica para Conjunto Independente Máximo

executam em tempo subexponencial para quase todo grafo

formalmente correlacionável com nobound

Programação Dinâmica

Chvátal [1977]: família de algoritmos de programação dinâmica para Conjunto Independente Máximo

executam em tempo subexponencial para quase todo grafo

formalmente correlacionável com nobound

resultado aplicável a nobound

Programação Dinâmica

Chvátal [1977]: família de algoritmos de programação dinâmica para Conjunto Independente Máximo

executam em tempo subexponencial para quase todo grafo

formalmente correlacionável com nobound

resultado aplicável a nobound

Teorema

$|T_{nobound}(G)|$ *é subexponencial (em* $|V(G)|$*) para quase todo* G

Programação Dinâmica

Chvátal [1977]: família de algoritmos de programação dinâmica para Conjunto Independente Máximo

executam em tempo subexponencial para quase todo grafo

formalmente correlacionável com nobound

resultado aplicável a nobound

Teorema

$|T_{nobound}(G)|$ *é subexponencial (em* $|V(G)|$*) para quase todo* G

$|T_{\mathcal{A}}(G)| < |T_{\text{nobound}}(G)|$, para todo algoritmo de B&B $\mathcal{A}$

Análise mais refinada

Análise mais refinada

Pittel [1982]

Análise mais refinada

Pittel [1982]: análise mais refinada da família de algoritmos de Chvátal [1977]

Análise mais refinada

Pittel [1982]: análise mais refinada da família de algoritmos de Chvátal [1977]

Teorema

o número de subproblemas gerados na execução dos algoritmos de Chvátal [1977] satisfaz (para quase todo G)

$$n^{(1/4-\varepsilon)\lg n} \leq s \leq n^{(1/2+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0,$$

Análise mais refinada

Pittel [1982]: análise mais refinada da família de algoritmos de Chvátal [1977]

Teorema

o número de subproblemas gerados na execução dos algoritmos de Chvátal [1977] satisfaz (para quase todo G)

$$n^{(1/4-\varepsilon)\lg n} \leq s \leq n^{(1/2+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0,$$

para quase todo grafo, estes algoritmos

Análise mais refinada

Pittel [1982]: análise mais refinada da família de algoritmos de Chvátal [1977]

Teorema

o número de subproblemas gerados na execução dos algoritmos de Chvátal [1977] satisfaz (para quase todo G)

$$n^{(1/4-\varepsilon)\lg n} \leq s \leq n^{(1/2+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0,$$

para quase todo grafo, estes algoritmos

- geram quantidade superpolinomial, mas quasipolinomial de subproblemas

Análise mais refinada

Pittel [1982]: análise mais refinada da família de algoritmos de Chvátal [1977]

Teorema

o número de subproblemas gerados na execução dos algoritmos de Chvátal [1977] satisfaz (para quase todo G)

$$n^{(1/4-\varepsilon)\lg n} \leq s \leq n^{(1/2+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0,$$

para quase todo grafo, estes algoritmos

- geram quantidade superpolinomial, mas quasipolinomial de subproblemas
- executam em tempo e espaço superpolinomial, mas quasipolinomial

Consequências para MCBB

Consequências para MCBB

Teorema: Os algoritmos MCBB tem tempo médio de execução

$$n^{(1+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0$$

Consequências para MCBB

Teorema: Os algoritmos MCBB tem tempo médio de execução

$$n^{(1+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0$$

Teorema: $\mathbb{E}\left[|T_{\text{nobound}}(G)|\right] = n^{(1+o(1))\frac{1}{2}\lg n}$

Consequências para MCBB

Teorema: Os algoritmos MCBB tem tempo médio de execução

$$n^{(1+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0$$

Teorema: $\mathbb{E}\left[|T_{\text{nobound}}(G)|\right] = n^{(1+o(1))\frac{1}{2}\lg n}$

$\mathcal{A}$

Consequências para MCBB

Teorema: Os algoritmos MCBB tem tempo médio de execução

$$n^{(1+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0$$

Teorema: $\mathbb{E}\left[|T_{\text{nobound}}(G)|\right] = n^{(1+o(1))\frac{1}{2}\lg n}$

$\mathcal{A}$: algoritmo de B&B para MaxClique

Consequências para MCBB

Teorema: Os algoritmos MCBB tem tempo médio de execução

$$n^{(1+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0$$

Teorema: $\mathbb{E}\left[|T_{\text{nobound}}(G)|\right] = n^{(1+o(1))\frac{1}{2}\lg n}$

$\mathcal{A}$: algoritmo de B&B para MaxClique $\implies |T_{\mathcal{A}}(G)| \leq |T_{\text{nobound}}(G)|$

Consequências para MCBB

Teorema: Os algoritmos MCBB tem tempo médio de execução

$$n^{(1+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0$$

Teorema: $\mathbb{E}\left[|T_{\text{nobound}}(G)|\right] = n^{(1+o(1))\frac{1}{2}\lg n}$

$\mathcal{A}$: algoritmo de B&B para MaxClique $\implies |T_{\mathcal{A}}(G)| \leq |T_{\text{nobound}}(G)|$

detalhes

Consequências para MCBB

Teorema: Os algoritmos MCBB tem tempo médio de execução

$$n^{(1+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0$$

Teorema: $\mathbb{E}\left[|T_{\text{nobound}}(G)|\right] = n^{(1+o(1))\frac{1}{2}\lg n}$

$\mathcal{A}$: algoritmo de B&B para MaxClique $\implies |T_{\mathcal{A}}(G)| \leq |T_{\text{nobound}}(G)|$

detalhes: Züge and Carmo [2018]

Consequências para MCBB

Teorema: Os algoritmos MCBB tem tempo médio de execução

$$n^{(1+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0$$

Teorema: $\mathbb{E}\left[|T_{\text{nobound}}(G)|\right] = n^{(1+o(1))\frac{1}{2}\lg n}$

$\mathcal{A}$: algoritmo de B&B para MaxClique $\implies |T_{\mathcal{A}}(G)| \leq |T_{\text{nobound}}(G)|$

detalhes: Züge and Carmo [2018]

todos os detalhes

Consequências para MCBB

Teorema: Os algoritmos MCBB tem tempo médio de execução

$$n^{(1+\varepsilon)\lg n}, \text{ para todo } \varepsilon > 0$$

Teorema: $\mathbb{E}\left[|T_{\text{nobound}}(G)|\right] = n^{(1+o(1))\frac{1}{2}\lg n}$

$\mathcal{A}$: algoritmo de B&B para MaxClique $\implies |T_{\mathcal{A}}(G)| \leq |T_{\text{nobound}}(G)|$

detalhes: Züge and Carmo [2018]

todos os detalhes: Züge [2017]

Aplicação na Metodologia Experimental

Aplicação na Metodologia Experimental

comparação via “benchmark”s limita conclusões

Aplicação na Metodologia Experimental

comparação via “benchmark”s limita conclusões: desestruturada

Aplicação na Metodologia Experimental

comparação via "benchmark"s limita conclusões: desestruturada

$$|T_{\mathcal{A}}(G)| \leq n^{c \lg n}, \quad \frac{1}{4} \leq c \leq \frac{1}{2}$$

Aplicação na Metodologia Experimental

comparação via "benchmark"s limita conclusões: desestruturada

$$|T_{\mathcal{A}}(G)| \leq n^{c \lg n}, \quad \frac{1}{4} \leq c \leq \frac{1}{2}$$

ideia: usar o valor do "menor c" como índice de efetividade do algoritmo

Aplicação na Metodologia Experimental

comparação via "benchmark"s limita conclusões: desestruturada

$$|T_{\mathcal{A}}(G)| \leq n^{c \lg n}, \quad \frac{1}{4} \leq c \leq \frac{1}{2}$$

ideia: usar o valor do "menor c" como índice de efetividade do algoritmo

$$\mathcal{R}_{\mathcal{A}}(G) := \frac{\lg |T_{\mathcal{A}}(G)|}{(\lg n)^2}$$

Aplicação na Metodologia Experimental

comparação via "benchmark"s limita conclusões: desestruturada

$$|T_{\mathcal{A}}(G)| \leq n^{c \lg n}, \quad \frac{1}{4} \leq c \leq \frac{1}{2}$$

ideia: usar o valor do "menor c" como índice de efetividade do algoritmo

$$\mathcal{R}_{\mathcal{A}}(G) := \frac{\lg |T_{\mathcal{A}}(G)|}{(\lg n)^2}$$

estimador de c: média de $\mathcal{R}_{\mathcal{A}}(G)$

Aplicação na Metodologia Experimental

comparação via "benchmark"s limita conclusões: desestruturada

$$|T_{\mathcal{A}}(G)| \leq n^{c \lg n}, \quad \frac{1}{4} \leq c \leq \frac{1}{2}$$

ideia: usar o valor do "menor c" como índice de efetividade do algoritmo

$$\mathcal{R}_{\mathcal{A}}(G) := \frac{\lg |T_{\mathcal{A}}(G)|}{(\lg n)^2}$$

estimador de c: média de $\mathcal{R}_{\mathcal{A}}(G)$
$\mathcal{R}_{\mathcal{A}}(G) < \mathcal{R}_{\mathcal{B}}(G) \rightarrow$ "$\mathcal{A}$ é melhor que $\mathcal{B}$"

Aplicação na Metodologia Experimental

comparação via "benchmark"s limita conclusões: desestruturada

$$|T_{\mathcal{A}}(G)| \leq n^{c \lg n}, \quad \frac{1}{4} \leq c \leq \frac{1}{2}$$

ideia: usar o valor do "menor c" como índice de efetividade do algoritmo

$$\mathcal{R}_{\mathcal{A}}(G) := \frac{\lg |T_{\mathcal{A}}(G)|}{(\lg n)^2}$$

estimador de c: média de $\mathcal{R}_{\mathcal{A}}(G)$

$\mathcal{R}_{\mathcal{A}}(G) < \mathcal{R}_{\mathcal{B}}(G) \rightarrow$ "$\mathcal{A}$ é melhor que $\mathcal{B}$"

$\mathcal{R}_{\mathcal{A}}(G) < \frac{1}{2} \rightarrow$ "$\mathcal{A}$ é melhor que nobound"

Aplicação na Metodologia Experimental

comparação via "benchmark"s limita conclusões: desestruturada

$$|T_{\mathcal{A}}(G)| \leq n^{c \lg n}, \quad \frac{1}{4} \leq c \leq \frac{1}{2}$$

ideia: usar o valor do "menor c" como índice de efetividade do algoritmo

$$\mathcal{R}_{\mathcal{A}}(G) := \frac{\lg |T_{\mathcal{A}}(G)|}{(\lg n)^2}$$

estimador de c: média de $\mathcal{R}_{\mathcal{A}}(G)$

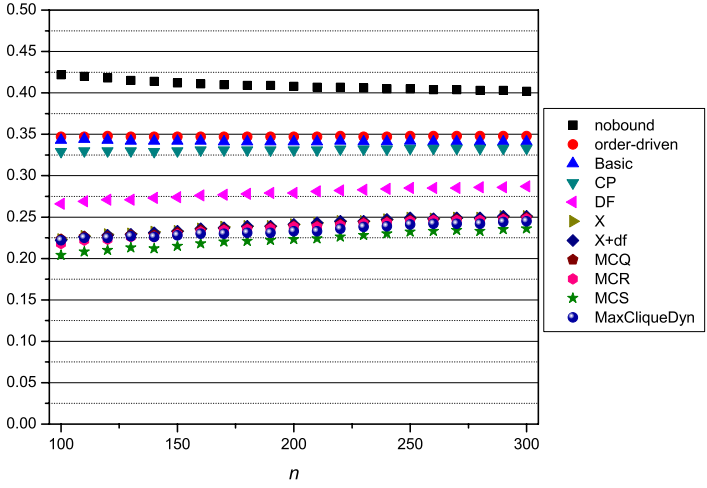
$\mathcal{R}_{\mathcal{A}}(G) < \mathcal{R}_{\mathcal{B}}(G) \rightarrow$ "$\mathcal{A}$ é melhor que $\mathcal{B}$"

$\mathcal{R}_{\mathcal{A}}(G) < \frac{1}{2} \rightarrow$ "$\mathcal{A}$ é melhor que nobound"

$\mathcal{R}_{\mathcal{A}}(G) < \frac{1}{4} \rightarrow$ "$\mathcal{A}$ é melhor que a família de Chvátal [1977]"

Avaliação

Avaliação


0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
100
150
200
250
300
n
nobound
order-driven
Basic
CP
DF
X
X+df
MCQ
MCR
MCS
MaxCliqueDyn

Em Números

Em Números

	$\mathcal{A}$	$R_{\mathcal{A}}(\mathcal{I})$	$\sigma_{R_{\mathcal{A}}(\mathcal{I})}$
1	MaxCLQ	0.218852	0.011797
2	BBMCX	0.221413	0.009996
3	MCS	0.222873	0.010820
4	MaxCliqueDyn	0.233902	0.008589
5	MCR	0.236429	0.010647
6	MCQ	0.237847	0.009628
7	$\chi + \mathrm{DF}$	0.240133	0.010251
8	χ	0.240332	0.009990
9	DF	0.279007	0.007480
10	CP	0.331465	0.003368
11	basic	0.341907	0.003361
12	order–driven	0.347358	0.002231
13	nobound	0.409352	0.006330

Em Números

	$\mathcal{A}$	$R_{\mathcal{A}}(\mathcal{I})$	$\sigma_{R_{\mathcal{A}}(\mathcal{I})}$
1	MaxCLQ	0.218852	0.011797
2	BBMCX	0.221413	0.009996
3	MCS	0.222873	0.010820
4	MaxCliqueDyn	0.233902	0.008589
5	MCR	0.236429	0.010647
6	MCQ	0.237847	0.009628
7	$\chi + \mathrm{DF}$	0.240133	0.010251
8	χ	0.240332	0.009990
9	DF	0.279007	0.007480
10	CP	0.331465	0.003368
11	basic	0.341907	0.003361
12	order–driven	0.347358	0.002231
13	nobound	0.409352	0.006330

$$n^{0.219 \lg n} \leq n^3 \colon\ n \leq 13295$$

Parametrização

Parametrização

CLIQUE é $W[1]$-completo para o tamanho da clique como parâmetro

Parametrização

CLIQUE é $W[1]$-completo para o tamanho da clique como parâmetro

G é d-degenerado

Parametrização

CLIQUE é $W[1]$-completo para o tamanho da clique como parâmetro

G é d-degenerado: todo subgrafo não vazio tem vértice de grau d

Parametrização

CLIQUE é $W[1]$-completo para o tamanho da clique como parâmetro

G é d-degenerado: todo subgrafo não vazio tem vértice de grau d

degeneração de G

Parametrização

CLIQUE é $W[1]$-completo para o tamanho da clique como parâmetro

G é d-degenerado: todo subgrafo não vazio tem vértice de grau d

degeneração de G: menor d tal que G é d-degenerado

Parametrização

CLIQUE é $W[1]$-completo para o tamanho da clique como parâmetro

G é d-degenerado: todo subgrafo não vazio tem vértice de grau d

degeneração de G: menor d tal que G é d-degenerado

Novacoski [2013]: CLIQUE é FPT quanto à degeneração

Parametrização

CLIQUE é $W[1]$-completo para o tamanho da clique como parâmetro

G é d-degenerado: todo subgrafo não vazio tem vértice de grau d

degeneração de G: menor d tal que G é d-degenerado

Novacoski [2013]: CLIQUE é FPT quanto à degeneração (LAWCG 2014)

Parametrização

CLIQUE é $W[1]$-completo para o tamanho da clique como parâmetro

G é d-degenerado: todo subgrafo não vazio tem vértice de grau d

degeneração de G: menor d tal que G é d-degenerado

Novacoski [2013]: CLIQUE é FPT quanto à degeneração (LAWCG 2014)

Walteros and Buchanan [2020]: instâncias "reais" tem baixa degeneração

Outras Tentativas

Outras Tentativas

Züge and Carmo [2016b] "retradução" de Li and Quan [2010b] (redução a MaxSat e otimização) em termos de Teoria dos Grafos

Outras Tentativas

Züge and Carmo [2016b] "retradução" de Li and Quan [2010b] (redução a MaxSat e otimização) em termos de Teoria dos Grafos

Züge et al. [2016]: comparação experimental com redução de MaxClique a MaxSat e uso do respectivo resolvedor

Outras Tentativas

Züge and Carmo [2016b] "retradução" de Li and Quan [2010b] (redução a MaxSat e otimização) em termos de Teoria dos Grafos

Züge et al. [2016]: comparação experimental com redução de MaxClique a MaxSat e uso do respectivo resolvedor

Liu [2017]: viabilidade de aplicação de ideias de "teste de propriedade" em algoritmos para MaxClique

Outras Tentativas

Züge and Carmo [2016b] "retradução" de Li and Quan [2010b] (redução a MaxSat e otimização) em termos de Teoria dos Grafos

Züge et al. [2016]: comparação experimental com redução de MaxClique a MaxSat e uso do respectivo resolvedor

Liu [2017]: viabilidade de aplicação de ideias de "teste de propriedade" em algoritmos para MaxClique

de Oliveira [2017]: exploração de outros limitantes, possibilidade de uso de algoritmo para CLIQUE como "bound" para MaxClique, etc

Outras Tentativas

Züge and Carmo [2016b] "retradução" de Li and Quan [2010b] (redução a MaxSat e otimização) em termos de Teoria dos Grafos

Züge et al. [2016]: comparação experimental com redução de MaxClique a MaxSat e uso do respectivo resolvedor

Liu [2017]: viabilidade de aplicação de ideias de "teste de propriedade" em algoritmos para MaxClique

de Oliveira [2017]: exploração de outros limitantes, possibilidade de uso de algoritmo para CLIQUE como "bound" para MaxClique, etc

Carmo et al. [2020]: pré-processamento via busca em largura lexicográfica

Outras Tentativas

Züge and Carmo [2016b] "retradução" de Li and Quan [2010b] (redução a MaxSat e otimização) em termos de Teoria dos Grafos

Züge et al. [2016]: comparação experimental com redução de MaxClique a MaxSat e uso do respectivo resolvedor

Liu [2017]: viabilidade de aplicação de ideias de "teste de propriedade" em algoritmos para MaxClique

de Oliveira [2017]: exploração de outros limitantes, possibilidade de uso de algoritmo para CLIQUE como "bound" para MaxClique, etc

Carmo et al. [2020]: pré-processamento via busca em largura lexicográfica

em andamento: implementação e avaliação experimental do algoritmo de Tarjan and Trojanowski [1976]

Em aberto

Em aberto

impressão

Em aberto

impressão: encontram clique máxima logo

Em aberto

impressão: encontram clique máxima logo maior parte do tempo gasto em certificar-se de que não há maior

Em aberto

impressão: encontram clique máxima logo maior parte do tempo gasto em certificar-se de que não há maior

maior parte do tempo gasto procurando certificado para instância de co-CLIQUE

Em aberto

impressão: encontram clique máxima logo maior parte do tempo gasto em certificar-se de que não há maior

maior parte do tempo gasto procurando certificado para instância de co-CLIQUE

algoritmos eficientes para co-CLIQUE?

Em aberto

impressão: encontram clique máxima logo maior parte do tempo gasto em certificar-se de que não há maior

maior parte do tempo gasto procurando certificado para instância de co-CLIQUE

algoritmos eficientes para co-CLIQUE?

estimativa de tempo de execução do algoritmo Inça and dos Reis [2017]

Resumo

1. algoritmos B&B para MaxClique são quasipolinomiais no caso médio

Resumo

1. algoritmos B&B para MaxClique são quasipolinomiais no caso médio

2. resultados analíticos $\times$ experimentais

Resumo

1. algoritmos B&B para MaxClique são quasipolinomiais no caso médio

2. resultados analíticos $\times$ experimentais: distância entre pior caso e caso médio

Resumo

1. algoritmos B&B para MaxClique são quasipolinomiais no caso médio

2. resultados analíticos $\times$ experimentais: distância entre pior caso e caso médio ("fenômeno quicksort")

Obrigado!

Referências I

B. Bollobás. **Random Graphs**. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 2001. ISBN 9780521797221.

Joshua Brakensiek, Marijn Heule, John Mackey, and David Narváez. The resolution of keller's conjecture. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, **Automated Reasoning**, pages 48–65, Cham, 2020. Springer International Publishing. ISBN 978-3-030-51074-9.

Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. **Commun. ACM**, 16(9):575–577, September 1973. ISSN 0001-0782. doi: http://dx.doi.org/10.1145/362342.362367. URL http://dx.doi.org/10.1145/362342.362367.

Referências II

Renato Carmo and Alexandre Züge. Branch and bound algorithms for the maximum clique problem under a unified framework. **Journal of the Brazilian Computer Society**, 18(2):137–151, December 2012. ISSN 0104-6500. doi: 10.1007/s13173-011-0050-6. URL https://link.springer.com/article/10.1007%2Fs13173-011-0050-6?LI=true.

Renato Carmo, Matheus Vinicius Correa, and Alexandre Prusch Züge. Lexicographic breadth first-search and branch and bound algorithms for the maximum clique problem. In **Proc. IX Latin American Workshop on Cliques in Graphs**, Goiânia, 2020.

Randy Carraghan and Panos M. Pardalos. An exact algorithm for the maximum clique problem. **Operations Research Letters**, 9(6):375–382, 1990. ISSN 0167-6377. doi: 10.1016/0167-6377(90)90057-C. URL http://dx.doi.org/10.1016/0167-6377(90)90057-C.

Referências III

Václav Chvátal. Determining the stability number of a graph. **SIAM Journal on Computing**, 6(4):643–662, December 1977. doi: 10.1137/0206046. URL http://dx.doi.org/10.1137/0206046.

Matheus Vinicius Correa. Lexbfs e algoritmos para busca em largura lexicográfica e algoritmos de solução exata para o problema da clique máxima. Mestrado, Programa de Pós-Graduação em Informática da UFPR, August 2020. Orientadores: Renato Carmo e Alexandre Züge.

Guilherme Bastos de Oliveira. Aceleração de algoritmos para o problema da clique máxima. Trabalho de Conclusão de Curso, 2017.

Cleverson Sebastião dos Anjos. Análise experimental de algoritmos. Master's thesis, Universidade Federal do Paraná, August 2015. URL http://acervodigital.ufpr.br/handle/1884/40875.

Referências IV

Cleverson Sebastiao dos Anjos, Alexandre Prusch Züge, and Renato Carmo. An experimental analysis of exact algorithms for the maximum clique problem. **Matemática Contemporânea**, 44:1–20, 2016. URL http://mc.sbm.org.br/wp-content/uploads/sites/15/2016/02/44-17.pdf.

Rodney G. Downey and Michael R. Fellows. **Parameterized Computational Feasibility**, chapter 7, pages 219–244. Birkhäuser Boston, Boston, MA, 1995. ISBN 978-1-4612-2566-9. doi: 10.1007/978-1-4612-2566-9_7. URL http://dx.doi.org/10.1007/978-1-4612-2566-9_7.

Elias P. Duarte Jr., Thiago Garrett, Luis C. E. Bona, Renato Carmo, and Alexandre P. Züge. Finding stable cliques of planetlab nodes. In **2010 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)**, pages 317–322. IEEE, June 2010. ISBN 978-1-4244-7500-1. doi: 10.1109/DSN.2010.5544300. URL http://dx.doi.org/10.1109/DSN.2010.5544300.

Referências V

Torsten Fahle. Simple and fast: Improving a branch-and-bound algorithm for maximum clique. In Rolf Möhring and Rajeev Raman, editors, **Proceedings of the 10th Annual European Symposium on Algorithms (ESA 2002)**, volume 2461 of **Lecture Notes in Computer Science**, pages 485–498, Berlin, Heidelberg, 2002. Springer. ISBN 978-3-540-45749-7. doi: 10.1007/3-540-45749-6_44. URL http://dx.doi.org/10.1007/3-540-45749-6_44.

Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. Measure and conquer: a simple $O(2^{0.288n})$ independent set algorithm. In **Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithms**, SODA, pages 18–25, Philadelphia, PA, USA, 2006. SIAM. ISBN 0-89871-605-5. doi: 10.1145/1109557.1109560. URL http://dx.doi.org/10.1145/1109557.1109560.

Referências VI

Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, Gregor Hendel, Christopher Hojny, Thorsten Koch, Pierre Le Bodic, Stephen J. Maher, Frederic Matter, Matthias Miltenberger, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Christine Tawfik, Stefan Vigerske, Fabian Wegscheider, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 7.0. ZIB-Report 20-10, Zuse Institute Berlin, March 2020. URL http://nbn-resolving.de/urn:nbn:de:0297-zib-78023.

William Gibson. **Johnny Mnemonic**. Ace Books, 1995.

Aric [Los Alamos National Laboratory] Hagberg, Pieter [Los Alamos National Laboratory] Swart, and Daniel [COLGATE UNIV] S Chult. Exploring network structure, dynamics, and function using networkx. In **SCIPY 08**, United States, Jan 2008. URL https://www.osti.gov/biblio/960616. Research Org.: Los Alamos National Lab. (LANL), Los Alamos, NM (United States); Sponsor Org.: USDOE.

Referências VII

Marijn Heule. Schur number five. In **Proceedings of the AAAI Conference on Artificial Intelligence**, volume 32, Apr. 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/12209.

Marijn J. H. Heule, Oliver Kullmann, and Victor W. Marek. Solving and verifying the boolean pythagorean triples problem via cube-and-conquer. In Nadia Creignou and Daniel Le Berre, editors, **Theory and Applications of Satisfiability Testing – SAT 2016**, pages 228–245, Cham, 2016. Springer International Publishing. ISBN 978-3-319-40970-2.

Boyi Hou, Zhuo Wang, Qun Chen, Bo Suo, Chao Fang, Zhanhuai Li, and Zachary G. Ives. Efficient maximal clique enumeration over graph data. **Data Science and Engineering**, 1(4):219–230, 2016. ISSN 2364-1541. doi: 10.1007/s41019-017-0033-5. URL http://dx.doi.org/10.1007/s41019-017-0033-5.

Referências VIII

Paulo Guilherme Inça and Rodrigo Camargo dos Reis. Estimativa on-line de tempo de execução para algoritmos de branch-and-bound. Trabalho de Conclusão de Curso, 2017.

Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, **Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department**, pages 85–103, Boston, MA, 1972. Springer US. ISBN 978-1-4684-2001-2. doi: 10.1007/978-1-4684-2001-2_9. URL https://doi.org/10.1007/978-1-4684-2001-2_9.

Referências IX

Janez Konc and Dušanka Janežič. An improved branch and bound algorithm for the maximum clique problem. **MATCH Communications in Mathematical and in Computer Chemistry**, June 2007. URL http://www.sicmm.org/konc/articles/match2007.pdf.

Nikolay Lavnikevich. On the complexity of maximum clique algorithms: usage of coloring heuristics leads to the $\Omega(2^{n/5})$ algorithm running time lower bound, 2013. URL http://arxiv.org/abs/1303.2546.

Chu-Min Li and Zhe Quan. An efficient branch-and-bound algorithm based on MaxSAT for the maximum clique problem. In **Twenty-Fourth Conference on Artificial Intelligence (AAAI)**, pages 128–133. AAAI Publications, 2010a. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1611.

Referências X

Chu-Min Li and Zhe Quan. Combining graph structure exploitation and propositional reasoning for the maximum clique problem. In **2010 22nd IEEE International Conference on Tools with Artificial Intelligence**, volume 1, pages 344–351. IEEE, 2010b. doi: 10.1109/ictai.2010.57. URL http://dx.doi.org/10.1109/ICTAI.2010.57.

Chu-Min Li, Zhiwen Fang, and Ke Xu. Combining MaxSAT reasoning and incremental upper bound for the maximum clique problem. In **25th International Conference on Tools with Artificial Intelligence (ICTAI)**, pages 939–946. IEEE, 2013. doi: 10.1109/ICTAI.2013.143. URL http://dx.doi.org/10.1109/ICTAI.2013.143.

Felix Yowtang Liu. Um algoritmo de teste de propriedades aplicado ao problema da clique. Trabalho de Conclusão de Curso, 2017.

Referências XI

Evgeny Maslov, Mikhail Batsyn, and Panos M. Pardalos. Speeding up mcs algorithm for the maximum clique problem with ils heuristic and other enhancements. In Boris I. Goldengorin, Valery A. Kalyagin, and Panos M. Pardalos, editors, **Models, Algorithms, and Technologies for Network Analysis: Proceedings of the Second International Conference on Network Analysis**, volume 59 of **Springer Proceedings in Mathematics & Statistics**, pages 93–99, New York, NY, USA, 2013. Springer. ISBN 978-1-4614-8588-9. doi: 10.1007/978-1-4614-8588-9_7. URL http://dx.doi.org/10.1007/978-1-4614-8588-9_7.

Catherine C McGeoch. **A guide to experimental algorithmics**. Cambridge University Press, 2012.

Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, ii. **Journal of Symbolic Computation**, 60:94–112, 2014. ISSN 0747-7171. doi: https://doi.org/10.1016/j.jsc.2013.09.003. URL https://www.sciencedirect.com/science/article/pii/S0747717113001193.

Referências XII

J. Moon and L. Moser. On cliques in graphs. **Israel Journal of Mathematics**, 3(1): 23–28, March 1965. doi: 10.1007/BF02760024. URL http://dx.doi.org/10.1007/BF02760024.

Kevin A. Naudé. Refined pivot selection for maximal clique enumeration in graphs. **Theoretical Computer Science**, 613:28–37, 2015. ISSN 0304-3975. doi: 10.1016/j.tcs.2015.11.016. URL http://dx.doi.org/10.1016/j.tcs.2015.11.016.

Jonilso Vianei Novacoski. Uma introdução à complexidade computacional parametrizada. Master's thesis, Universidade Federal do Paraná, 2013.

Patric R. J. Östergård. A fast algorithm for the maximum clique problem. **Discrete Applied Mathematics**, 120(1):197–207, August 2002. doi: 10.1016/S0166-218X(01)00290-6. URL http://dx.doi.org/10.1016/S0166-218X(01)00290-6.

Referências XIII

Boris Pittel. On the probable behaviour of some algorithms for finding the stability number of a graph. **Mathematical Proceedings of the Cambridge Philosophical Society**, 92:511–526, November 1982. ISSN 1469-8064. doi: 10.1017/S0305004100060205. URL http://dx.doi.org/10.1017/S0305004100060205.

Patrick Prosser. Exact algorithms for maximum clique: A computational study. **Algorithms**, 5(4):545–587, 2012. ISSN 1999-4893. doi: 10.3390/a5040545. URL http://www.mdpi.com/1999-4893/5/4/545.

J. M. Robson. Finding a maximum independent set in time $O(2^{n/4})$. Technical report, Technical Report 1251-01, LaBRI, Université de Bordeaux I, Université de Bordeaux, January 2001. URL http://www.labri.fr/perso/robson/mis/techrep.html.

Pablo San Segundo and Cristobal Tapia. Relaxed approximate coloring in exact maximum clique search. **Computers & Operations Research**, 44:185–192, 2014. ISSN 0305-0548. doi: 10.1016/j.cor.2013.10.018. URL http://dx.doi.org/10.1016/j.cor.2013.10.018.

Referências XIV

Pablo San Segundo, Diego Rodríguez-Losada, and Agustín Jiménez. An exact bit-parallel algorithm for the maximum clique problem. **Computers & Operations Research**, 38(2):571–581, 2011.

Pablo San Segundo, Jorge Artieda, Rafael Leon, and Cristobal Tapia. An enhanced infra-chromatic bound for the maximum clique problem. In Panos M. Pardalos, Piero Conca, Giovanni Giuffrida, and Giuseppe Nicosia, editors, **Machine Learning, Optimization, and Big Data: Second International Workshop (MOD 2016)**, volume 10122 of **Lecture Notes in Computer Science**, pages 306–316, Cham, 2016a. Springer International Publishing. ISBN 978-3-319-51469-7. doi: 10.1007/978-3-319-51469-7_26. URL http://dx.doi.org/10.1007/978-3-319-51469-7_26.

Referências XV

Pablo San Segundo, Alvaro Lopez, Mikhail Batsyn, Alexey Nikolaev, and Panos M. Pardalos. Improved initial vertex ordering for exact maximum clique search. **Applied Intelligence**, 45(3):868–880, 2016b. ISSN 1573-7497. doi: 10.1007/s10489-016-0796-9. URL http://dx.doi.org/10.1007/s10489-016-0796-9.

Pablo San Segundo, Jorge Artieda, Mikhail Batsyn, and Panos M. Pardalos. An enhanced bitstring encoding for exact maximum clique search in sparse graphs. **Optimization Methods and Software**, 32(2):312–335, 2017. doi: 10.1080/10556788.2017.1281924. URL http://dx.doi.org/10.1080/10556788.2017.1281924.

Steven S Skiena. **The algorithm design manual**. Springer International Publishing, 2020.

Referências XVI

William Stein and David Joyner. SAGE: System for algebra and geometry experimentation. **SIGSAM Bulletin (ACM Special Interest Group on Symbolic and Algebraic Manipulation)**, 39(2):61–64, June 2005. ISSN 0163-5824 (print), 1557-9492 (electronic).

Robert E. Tarjan and Anthony E. Trojanowski. Finding a maximum independent set. Technical report, Computer Science Department, School of Humanities and Sciences, Stanford University, Stanford, CA, USA, 1976.

E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. **Theoretical Computer Science**, 363(1):28–42, October 2006. ISSN 03043975. doi: 10.1016/j.tcs.2006.06.015. URL http://dx.doi.org/10.1016/j.tcs.2006.06.015.

Referências XVII

Etsuji Tomita and Toshikatsu Kameda. An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments. **Journal of Global Optimization**, 37(1):95–111, January 2007. ISSN 0925-5001. doi: http://dx.doi.org/10.1007/s10898-006-9039-7. URL http://dx.doi.org/10.1007/s10898-006-9039-7.

Etsuji Tomita and Tomokazu Seki. An efficient branch-and-bound algorithm for finding a maximum clique. In Cristian S. Calude, Michael J. Dinneen, and Vincent Vajnovszki, editors, **Proceedings of the 4th International Conference on Discrete Mathematics and Theoretical Computer Science (DMTCS)**, volume 2731 of **Lecture Notes in Computer Science**, pages 278–289. Springer, Berlin, Heidelberg, 2003. ISBN 978-3-540-40505-4. doi: 10.1007/3-540-45066-1_22. URL http://dx.doi.org/10.1007/3-540-45066-1_22.

Referências XVIII

Etsuji Tomita, Yoichi Sutani, Takanori Higashi, Shinya Takahashi, and Mitsuo Wakatsuki. A simple and faster branch-and-bound algorithm for finding a maximum clique. In Md Rahman and Satoshi Fujita, editors, **Proceedings of the 4th International Workshop Algorithms and Computation (WALCOM)**, volume 5942 of **Lecture Notes in Computer Science**, pages 191–203. Springer, Berlin, Heidelberg, 2010. ISBN 978-3-642-11439-7. doi: 10.1007/978-3-642-11440-3_18. URL http://dx.doi.org/10.1007/978-3-642-11440-3_18.

Etsuji Tomita, Kohei Yoshida, Takuro Hatta, Atsuki Nagao, Hiro Ito, and Mitsuo Wakatsuki. A much faster branch-and-bound algorithm for finding a maximum clique. In Daming Zhu and Sergey Bereg, editors, **Proceedings of the 10th International Workshop Frontiers in Algorithmics (FAW)**, volume 9711 of **Lecture Notes in Computer Science**, pages 215–226, Cham, 2016. Springer International Publishing. ISBN 978-3-319-39817-4. doi: 10.1007/978-3-319-39817-4_21. URL http://dx.doi.org/10.1007/978-3-319-39817-4_21.

Referências XIX

Rim van Wersch and Steven Kelk. Toto: An open database for computation, storage and retrieval of tree decompositions. **Discrete Applied Mathematics**, 217: 389–393, 2017. ISSN 0166-218X. doi: https://doi.org/10.1016/j.dam.2016.09.023. URL https://www.sciencedirect.com/science/article/pii/S0166218X16304164.

Jose L. Walteros and Austin Buchanan. Why is maximum clique often easy in practice? **Operations Research**, 0(0):null, 2020. doi: 10.1287/opre.2019.1970. URL https://doi.org/10.1287/opre.2019.1970.

Mingyu Xiao and Hiroshi Nagamochi. Exact algorithms for maximum independent set. **Information and Computation**, 255(Part 1):126–146, 2017. ISSN 0890-5401. doi: https://doi.org/10.1016/j.ic.2017.06.001. URL http://www.sciencedirect.com/science/article/pii/S0890540117300950.

Emre Yolcu, Scott Aaronson, and Marijn J. H. Heule. An automated approach to the collatz conjecture, 2021.

Referências XX

David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In **Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing**, STOC, pages 681–690, New York, NY, USA, 2006. ACM. ISBN 1-59593-134-1. doi: 10.1145/1132516.1132612. URL http://doi.acm.org/10.1145/1132516.1132612.

Alexandre Züge and Renato Carmo. Worst case instances for maximum clique algorithms and how to avoid them. In **Abstracts – Latin American Workshop on Cliques in Graphs**, page 24, La Plata, Argentina, 2016a. Universidad Nacional de La Plata.

Alexandre Züge, Renato Carmo, Ricardo Tavares de Oliveira, and Fabiano Silva. Using pmaxsat techniques to solve the maximum clique problem. In **Abstracts – Latin American Workshop on Cliques in Graphs**, page 25, La Plata, Argentina, 2016. Universidad Nacional de La Plata.

Referências XXI

Alexandre Prusch Züge. Solução exata do problema da clique máxima. Master's thesis, Universidade Federal do Paraná, nov 2011. URL http://dspace.c3sl.ufpr.br/dspace/handle/1884/27588.

Alexandre Prusch Züge and Renato Carmo. Maximum clique via maxsat and back again. **Matemática Contemporânea**, 44:1–10, 2016b. URL http://mc.sbm.org.br/wp-content/uploads/sites/15/2016/02/44-18.pdf.

Alexandre Züge. **Algoritmos para o Problema da Clique Máxima: análise e comparação experimental**. PhD thesis, Universidade Federal do Paraná, sep 2017. 28/9/2017.

Alexandre Prusch Züge and Renato Carmo. On comparing algorithms for the maximum clique problem. **Discrete Applied Mathematics**, 247:1–13, 2018. ISSN 0166-218X. doi: https://doi.org/10.1016/j.dam.2018.01.005. URL http://www.sciencedirect.com/science/article/pii/S0166218X18300167.