Problemas NP-Completo

Última alteração: 28 de setembro de 2011

Problemas NP-Completo

- Ainda não foi descoberto nenhum algoritmo de tempo polinomial para um problema NP-Completo, nem ninguém ainda foi capaz de provar que não pode existir nenhum algoritmo de tempo polinomial para qualquer deles
- Essa questão chamada $P \neq NP$ foi um dos mais profundos e surpreendentes problemas de pesquisa abertos na Ciência da Computação teórica desde que foi proposto pela primeira vez em 1971

Caráter NP-Completo e as Classes P e NP

- Classe P
  - Consiste nos problemas que podem ser resolvidos em tempo polinomial
- Classe NP
  - Consiste nos problemas que são verificáveis em tempo polinomial
  - Se tivéssemos de algum modo um “certificado” de uma solução, poderíamos verificar se o certificado é correto em tempo polinomial no tamanho da entrada para o problema

Caráter NP-Completo e as Classes P e NP

- Classe NP
  - Por exemplo, no problema do ciclo hamiltoniano, dado um grafo orientado $G = (V, A)$, um certificado seria uma seqüência $\langle v_1, v_2, ..., v_{|V|} \rangle$ de $|V|$ vértices
  - É fácil verificar em tempo polinomial que
    - $(v_i, v_{i+1}) \in A$ para $i = 1, 2, ..., |V| - 1$
    - $(v_{|V|}, v_1) \in A$
  - Como outro exemplo, no problema da satisfabilidade, um certificado seria uma atribuição de valores a variáveis
  - Podemos verificar facilmente em tempo polinomial que essa atribuição satisfaz à fórmula booleana

Caráter NP-Completo e as Classes P e NP

- Qualquer problema em P também está em NP
  - Se um problema está em P então podemos resolvê-lo em tempo polinomial sem sequer receber um certificado
  - Vamos formalizar essa noção mais adiante
  - No momento, podemos acreditar que $P \subseteq NP$
- A questão aberta é se P é ou não um subconjunto próprio de NP

Caráter NP-Completo e as Classes P e NP

- Classe NP-Completo
  - Informalmente, um problema está na classe NP-Completo se ele está em NP e é tão “difícil” quanto qualquer problema em NP
  - Definiremos formalmente o que significa ser tão difícil quanto qualquer problema em NP mais adiante
  - Vamos declarar sem provar que, se *qualquer* problema NP-Completo pode ser resolvido em tempo polinomial, então *todo* problema NP-Completo tem um algoritmo de tempo polinomial

Caráter NP-Completo e as Classes P e NP

- Classe NP-Completo
  - A maioria dos teóricos acredita que os problemas NP-Completos são intratáveis pois, dada a ampla faixa de problemas NP-Completos que foram estudados até hoje, sem qualquer progresso em direção a uma solução de tempo polinomial, seria verdadeiramente espantoso se todos eles pudessem ser resolvidos em tempo polinomial
  - Ainda assim, dado o esforço dedicado há tanto tempo para provar que os problemas NP-Completos são intratáveis – sem um resultado conclusivo – não podemos eliminar a possibilidade de que os problemas NP-Completos possam de fato ser resolvidos em tempo polinomial

Caráter NP-Completo e as Classes P e NP

- Para se tornar um bom projetista de algoritmos, você deve entender os rudimentos da teoria dos problemas NP-Completos
- Se puder estabelecer um problema como NP-Completo, você fornecerá uma boa evidência de sua intratabilidade
- Como um engenheiro, você faria melhor gastando seu tempo no desenvolvimento de um algoritmo de aproximação ou resolvendo um caso especial tratável, em vez de procurar por um algoritmo rápido que resolva o problema exatamente
- Além disso, muitos problemas naturais e interessantes que na superfície não parecem mais difíceis que a ordenação ou a pesquisa de grafos são de fato NP-Completos

Problemas de Decisão *Versus* Problemas de Otimização

- Muitos problemas de interesse são **problemas de otimização**, em que cada solução possível tem um valor associado, e desejamos encontrar a solução possível com o melhor valor
  - Por exemplo, no problema SHORTEST-PATH, temos um grafo não orientado *G* e vértices *u* e *v*, e desejamos encontrar o caminho *u* até *v* que utiliza o menor número de arestas
- O caráter NP-Completo não se aplica diretamente a problemas de otimização, mas a **problemas de decisão**, em que a resposta é simplesmente “sim” ou “não”

Problemas de Decisão *Versus* Problemas de Otimização

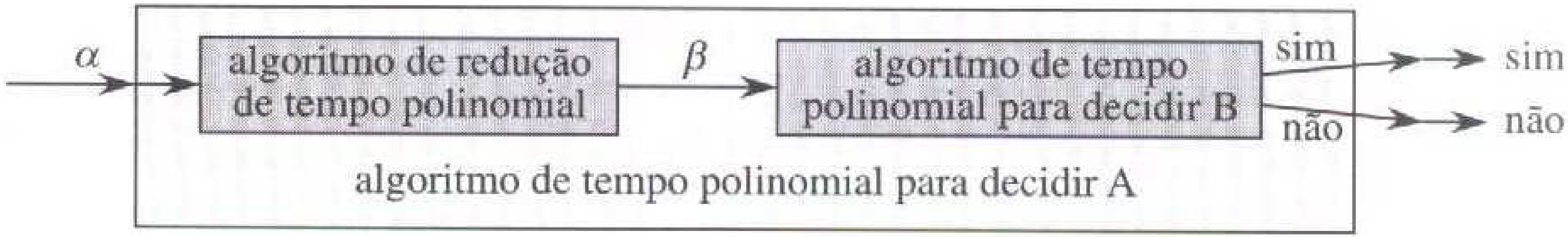
- Existe um relacionamento conveniente entre problemas de otimização e de decisão
- Podemos formular um problema de otimização como um problema de decisão relacionado impondo um limite sobre o valor a ser otimizado
  - Por exemplo, no caso de SHORTEST-PATH, um problema de decisão relacionado, que chamamos PATH, é se, dado um grafo orientado *G*, vértices *u* e *v*, e um inteiro *k*, existe um caminho de *u* até *v* consistindo em no máximo *k* arestas

Problemas de Decisão *Versus* Problemas de Otimização

- O problema de decisão é de certo modo “mais fácil” ou, pelo menos, “não mais difícil” que o problema de otimização
  - Por exemplo, podemos resolver PATH resolvendo SHORTEST-PATH, e depois comparando o número de arestas no caminho mais curto encontrado ao valor do parâmetro de problema de decisão *k*
- Se um problema de otimização é fácil, seu problema de decisão relacionado também é fácil
- Se pudermos fornecer evidências de que um problema de decisão é difícil, também forneceremos evidências de que seu problema de otimização relacionado é difícil
- Embora restrinja a atenção a problemas de decisão, a teoria de problemas NP-Completos freqüentemente também tem implicações para problemas de otimização

Reduções

- Considere um problema de decisão *A* que gostaríamos de resolver em tempo polinomial
- Suponha que exista um problema de decisão diferente *B* que já sabemos como resolver em tempo polinomial
- Suponha que temos um procedimento que transforma qualquer instância *α* de *A* em alguma instância *β* de *B* com as seguintes características
  – A transformação demora tempo polinomial
  – As respostas são as mesmas, isto é, a resposta para *α* é "sim" se e somente se a resposta para *β* também é "sim"



Reduções

- O algoritmo de redução de tempo polinomial nos oferece um meio para resolver *A* em tempo polinomial
    1. Dada uma instância α de *A*, use um algoritmo de redução de tempo polinomial para transformá-la em uma instância β de *B*
    2. Execute o algoritmo de decisão de tempo polinomial para *B* sobre a instância β
    3. Use a resposta de β como a resposta para α
- Como cada uma dessas etapas demora tempo polinomial, temos um modo de decidir sobre α em tempo polinomial
- "Reduzindo" a solução de *A* à solução de *B*, usamos a "facilidade" de *B* para provar a "facilidade" de *A*

Reduções

- As reduções de tempo polinomial nos dão uma poderosa ferramenta para provar que diversos problemas pertencem a P
- Lema 34.3
  – Se $\pi_1 \propto \pi_2$, então $\pi_2 \in$ P implica $\pi_1 \in$ P

Reduções

- O caráter NP-Completo consiste em mostrar o quanto um problema é difícil, em vez de mostrar o quanto ele é fácil
- Usamos reduções de tempo polinomial da maneira oposta para mostrar que um problema é NP-Completo
- Vamos mostrar como podemos usar reduções de tempo polinomial para demonstrar que não pode existir nenhum algoritmo de tempo polinomial para um determinado problema *B*

Reduções

- Suponha que temos um problema de decisão *A* para o qual já sabemos que não existe nenhum algoritmo de tempo polinomial
- Suponha ainda que temos uma redução de tempo polinomial transformando instâncias de *A* em instâncias de *B*
- Podemos usar uma prova simples por contradição para mostrar que não é possível existir nenhum algoritmo de tempo polinomial para *B*
- Suponha que *B* tenha um algoritmo de tempo polinomial
- Usando o método de redução, teríamos um modo de resolver *A* em tempo polinomial, o que contradiz nossa hipótese de que não existe nenhum algoritmo de tempo polinomial para *A*

Reduções

- No caso do caráter NP-Completo, não podemos supor que não exista absolutamente nenhum algoritmo de tempo polinomial para o problema A
- A metodologia da prova é semelhante, no sentido de que provamos que o problema B é NP-Completo na hipótese de que o problema A também seja NP-Completo

Problemas NP-Completo

- Um problema de decisão π é NP-Completo se
  1. $\pi \in$ NP, e
  2. $\pi' \propto \pi$ para todo $\pi' \in$ NP
- Se um problema π satisfaz à propriedade 2, mas não necessariamente à propriedade 1, dizemos que π é NP-difícil

Problemas NP-Completo

- O caráter NP-Completo é o ponto crucial de se determinar se P é de fato igual a NP
- Teorema 34.4
  1. Se existir algum problema NP-Completo que possa ser resolvido em tempo polinomial, então P = NP.
  2. Se existir algum problema em NP que não possa ser resolvido em tempo polinomial, então nenhum problema NP-Completo pode ser resolvido em tempo polinomial

Problemas NP-Completo

- Teorema 34.4
  1. Se existir algum problema NP-Completo que possa ser resolvido em tempo polinomial, então P = NP.
- Prova
  – Suponha, por hipótese, que $\pi \in$ P e $\pi \in$ NP-Completo
  – Pela propriedade 2 da definição do caráter NP-Completo, $\pi' \propto \pi$ para todo $\pi' \in$ NP
  – Pelo Lema 34.3, se $\pi' \propto \pi$ então $\pi \in$ P implica $\pi' \in$ P. Então, também temos que $\pi' \in$ P para todo $\pi' \in$ NP
  – Portanto, se $\pi \in$ P e $\pi \in$ NP-Completo, então P = NP

Problemas NP-Completo

- Teorema 34.4

  2. Se existir algum problema em NP que não possa ser resolvido em tempo polinomial, então nenhum problema NP-Completo pode ser resolvido em tempo polinomial

- Prova
  - Suponha, por hipótese, que $\pi \notin$ P e $\pi \in$ NP
  - Suponha, por absurdo, que $\pi' \in$ NP-Completo e $\pi' \in$ P
  - Pela propriedade 2 da definição do caráter NP-Completo, $\pi'' \propto \pi'$ para todo $\pi'' \in$ NP
  - Pelo Lema 34.3, se $\pi'' \propto \pi'$ então $\pi' \in$ P implica $\pi'' \in$ P. Então, também temos que $\pi'' \in$ P para todo $\pi'' \in$ NP, o que contradiz a hipótese
  - Portanto, se $\pi \notin$ P e $\pi \in$ NP, então $\pi' \notin$ P para todo $\pi' \in$ NP-Completo

Problemas NP-Completo

- É por essa razão que a pesquisa sobre a questão P $\neq$ NP se concentra em torno dos problemas NP-Completos
- A maioria dos cientistas da computação teórica acredita que P $\neq$ NP
- Porém, por tudo que sabemos, alguém poderia apresentar um algoritmo de tempo polinomial para um problema NP-Completo, e desse modo provar que P = NP
- Não obstante, como ainda não foi descoberto nenhum algoritmo de tempo polinomial para qualquer problema NP-Completo, uma prova de que um problema

Provas do Caráter NP-Completo

- O caráter NP-Completo do problema da satisfabilidade de circuitos (CIRCUIT-SAT) se baseia em uma prova direta de que $\pi \propto$ CIRCUIT-SAT para todo problema $\pi \in$ NP
- Mostraremos como provar que os problemas são NP-Completo sem reduzir diretamente *todo* problema em NP ao problema dado

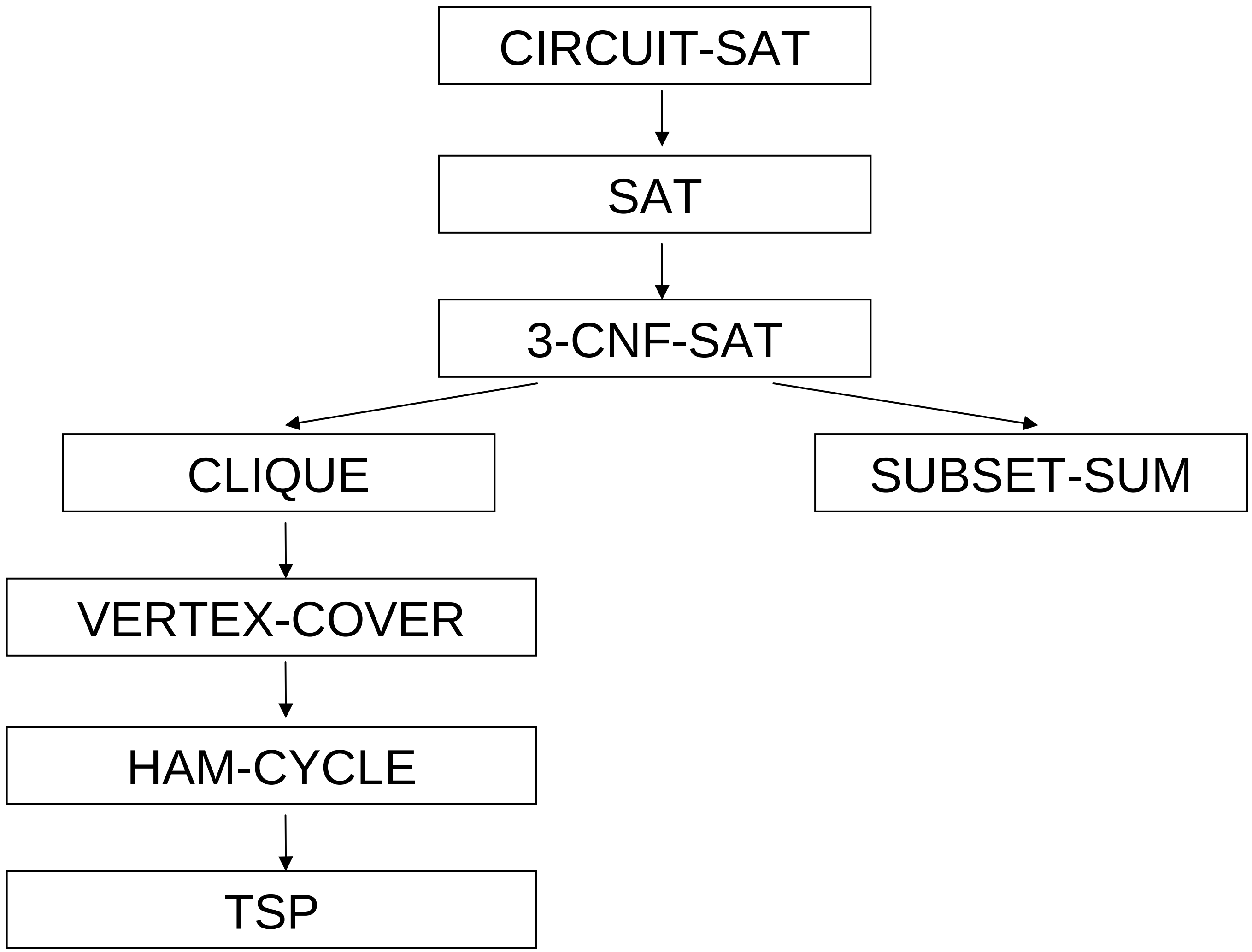
Provas do Caráter NP-Completo

- Lema 34.8
  - Se π é um problema tal que $\pi' \propto \pi$ para algum $\pi' \in$ NP-Completo, então π é NP-difícil
  - Além disso, se $\pi \in$ NP, então $\pi \in$ NP-Completo
- Prova
  - Tendo em vista que π' é NP-Completo, pela propriedade 2 da definição do caráter NP-Completo, $\pi'' \propto \pi'$ para todo $\pi'' \in$ NP
  - Por hipótese, $\pi' \propto \pi$ e, por transitividade, temos $\pi'' \propto \pi$, o que mostra que π é NP-difícil
  - Se $\pi \in$ NP, também temos $\pi \in$ NP-Completo
- Reduzindo a π um problema NP-Completo π' conhecido, reduzimos implicitamente a π todo problema em NP

Provas do Caráter NP-Completo

- Para provar que um problema π é NP-Completo
  - Prove que $\pi \in$ NP
  - Mostre que um problema NP-Completo conhecido π' pode ser polinomialmente transformado para π
- Saber que o problema da satisfabilidade de circuitos é NP-Completo nos permite provar muito mais facilmente que outros problemas são NP-Completos
- Além disso, à medida que desenvolvermos um catálogo de problemas NP-Completos conhecidos, teremos cada vez mais opções de problemas a partir dos quais poderá ser feita a redução

Estrutura das Provas do Caráter NP-Completo



Referência

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Algoritmos: Teoria e Prática. Campus, Tradução da Segunda Edição Americana, 2002.
  - Capítulo 34