Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação

Algoritmos Aproximativos

Lista de exercícios 1

1. Obtenha um algoritmo polinomial (1/2)-aproximativo para o problema do subgrafo acíclico: dado um grafo direcionado G, obtenha o maior número de arcos de G sem formar um ciclo direcionado. Prove que o fator de aproximação é 2. Encontre um exemplo que mostre que o fator de aproximação do seu algoritmo não pode ser melhorado. **Dica:** considerando que os vértices são rotulados com números inteiros, observe o conjunto das arestas de um vértice menor para um maior e o conjunto das arestas de um vértice maior para um menor.

2. Já vimos que o algoritmo Escalonamento-Graham tem fator de aproximação 2. Prove que esse fator de aproximação pode ser melhorado para $2 - \frac{1}{m}$, onde m é o número de máquinas. Para cada m, mostre uma instância com exatamente esse fator de aproximação no algoritmo.

3. Já vimos o algoritmo MinCC-Chvátal para o Problema da Cobertura Mínima por Conjuntos. O algoritmo visto é recursivo. Escreva uma versão não-recursiva desse algoritmo.

4. Considere o seguinte problema MaxCC$(U, w, \mathcal{S}, k)$ da Cobertura máxima (*maximum coverage problem*): dados um conjunto universo $U = \{u_1, \ldots, u_n\}$ finito, um peso racional $w_u, \forall u \in U$, uma coleção $\mathcal{S} = \{S_1, \ldots, S_m\}$ de subconjuntos de U e um inteiro positivo k, encontrar uma subcoleção $\mathcal{S}^*$ com k conjuntos de $\mathcal{S}$ de peso máximo. Este problema é NP-difícil [Garey, Johnson'79]. Mostre que o algoritmo guloso que escolhe a cada passo um conjunto de maior peso entre os elementos ainda não cobertos é $(1-(1-k^{-1})^k)$-aproximativo polinomial. Como $1-(1-k^{-1})^k > 1-1/e$, conclua que esse algoritmo é 0.63-aproximativo para o MaxCC.

5. O algoritmo guloso (exato e polinomial) do Problema da Mochila Fracionária, escolhe um a um os itens com maior razão custo-benefício $valor/peso$ até preencher a mochila ou terminarem os itens. Mostre que esse algoritmo não tem fator de aproximação constante para o Problema da Mochila 0-1 (booleana), ou seja, exiba uma família de instâncias para as quais esse algoritmo produz um fator de aproximação ε para qualquer $0 < \varepsilon < 1$.

6. Considere as três seguintes variantes do TSPM (Caixeiro Viajante Métrico). Na primeira, buscamos um caminho hamiltoniano (ao invés de ciclo hamiltoniano) de custo mínimo no grafo dado. Na segunda, queremos um caminho hamiltoniano de custo mínimo iniciando em um dado vértice s. Na terceira, queremos um caminho hamiltoniano de custo mínimo entre dois vértices dados s e t. Modifique o algoritmo de Christofides para que ele tenha fator de aproximação menor que 2 para cada uma dessas variantes.

7. Mostre que os algoritmos TSPM-RSL e TSPM-Christofides podem produzir péssimos resultados se aplicados a instâncias do TSP (Caixeiro Viajante) que não satisfazem a desigualdade triangular.

8. Considere o seguinte problema MaxCut(G, w) do Corte Máximo: dado um grafo G com peso racional positivo w_e em cada aresta e, encontre uma partição (V_1, V_2) do conjunto de vértices tal que a soma dos pesos das arestas entre V_1 e V_2 seja máxima. Prove que o seguinte algoritmo probabilístico tem razão 0.5 de aproximação esperada: para cada vértice v, colocar v em V_i, onde i é escolhido de modo aleatório e uniforme entre 1 e 2.

9. Escreva uma versão desaleatorizada do algoritmo 0.5-aproximativo da questão anterior para o Problema do Corte Máximo.

10. Considere a seguinte variação Max-k-Cut(G, w) do Problema do Corte Máximo: dado um grafo G com peso racional positivo w_e em cada aresta e, encontre uma partição $(V_1, V_2, \ldots, V_k)$ do conjunto de vértices tal que a soma dos pesos de cada aresta entre duas partes distintas V_i e V_j seja máxima. Prove que o seguinte algoritmo probabilístico tem razão $1 - \frac{1}{k}$ de aproximação esperada: Para cada vértice v, colocar v em V_i, onde i é escolhido de modo aleatório e uniforme em $\{1, \ldots, k\}$.

11. Prove que o seguinte algoritmo é uma 0.75-aproximação probabilística para MaxSat: com probabilidade $1/2$, execute um dos algoritmos aproximativos probabilísticos MaxSat-Johnson ou MaxSat-GW, retornando a valoração. A diferença desse algoritmo para o algoritmo MaxSat-Combinado-GW visto em aula é que, nesse último, executam-se os dois algoritmos escolhendo a valoração com maior número de cláusulas satisfeitas. **Dica:** Siga a demonstração do fator de aproximação esperado 0.75 do algoritmo MaxSat-Combinado-GW visto em aula.

12. Na aula sobre o algoritmo aproximativo MaxSat-GW (de Goemans e Willimason) para MaxSat, foi falado que era possível desaleatorizar este algoritmo. Escreva esta versão desaleatorizada do algoritmo e prove que possui a mesma razão de aproximação $1 - (1 - 1/k)^k$ do algoritmo probabilístico, quando cada cláusula tem no máximo k literais. Use-a para obter um algoritmo combinado desaleatorizado com fator de aproximação 0.75 e prove esta razão de aproximação. **Dicas:** Reveja o vídeo da aula; método das esperanças condicionais.

13. Explique rapidamente porque os problemas Escalonamento, Bin Packing, MaxCut, Max-k-Cut, MaxSat, Mochila, MinCC, MinCV, TSPM e TSP estão na classe NPO de problemas de otimização.

14. Dado um problema Π de otimização, seja Π_D a sua versão de decisão. Ou seja, dada uma instância I de Π e um racional k, $\Pi_D(\mathrm{I}, k)$ é SIM se existe uma solução de I com valor $\leq k$ (se min.) ou $\geq k$ (se max.). Caso contrário, $\Pi_D(\mathrm{I}, k)$ é NÃO. Prove que, se $\Pi_D(\mathrm{I}, k)$ é NP-Completo para algum k fixo, então não existe algoritmo polinomial para Π com fator de aproximação melhor que $1 \pm \frac{1}{k}$, a menos que $P = NP$. **Dica:** Comece com a versão de Π em que os valores das soluções são inteiros. **Observação:** Essa é uma das formas da técnica do Gap, que já usamos para provar inaproximabilidade do Bin-Packing.