Complexidade Parametrizada

Buscamos algoritmos cuja complexidade exponencial depende apenas de certos aspectos (parâmetros) da entrada.

Algoritmos tratáveis por parâmetro fixo - FPT

Seja Π um problema NP-difícil e seja $S = \{a_1, \dots, a_\ell\}$ um conjunto de aspectos de Π.

Chamamos de $\Pi(a_1, \dots, a_\ell)$ ou $\Pi(S)$ a versão parametrizada de Π onde os aspectos em S são parâmetros fixados

Um problema parametrizado $\Pi(S)$ pertence à classe XP se existe um algoritmo para resolver $\Pi(S)$ de tempo $f(S)\, n^{g(S)}$, onde n é o tamanho da entrada.

Um problema parametrizado $\Pi(S)$ pertence à classe FPT, ou é tratável por parâmetro fixo, se existe um algoritmo para resolver $\Pi(S)$ de tempo $f(S)\, n^c$, onde n é o tamanho da entrada e c é uma constante.

Clique (k)

Instância: um grafo $G=(V,E)$

Parâmetro: um inteiro positivo k.

Questão: G possui conjunto $S \subseteq V$ com $|S| \geq k$ tal que os vértices de S são mutuamente adjacentes?

Clique (k) admite um algoritmo de tempo $f(k) \cdot n^{g(k)}$, está em XP

Basta examinar para cada um dos $\binom{n}{k}$ subconjuntos de vértices, em tempo k^2, se os vértices são mutuamente adjacentes.

Cobertura por vértices (k)

Instância: um grafo $G=(V,E)$

Parâmetro: um inteiro positivo k.

Questão: G possui conjunto $I \subseteq V$ com $|I| \leq k$ tal que toda aresta $e \in E$ tem pelo menos um extremo em I?

Cobertura por vértices (k) admite algoritmo de tempo $f(k) \cdot n^c$, está em FPT.

Algoritmo 1

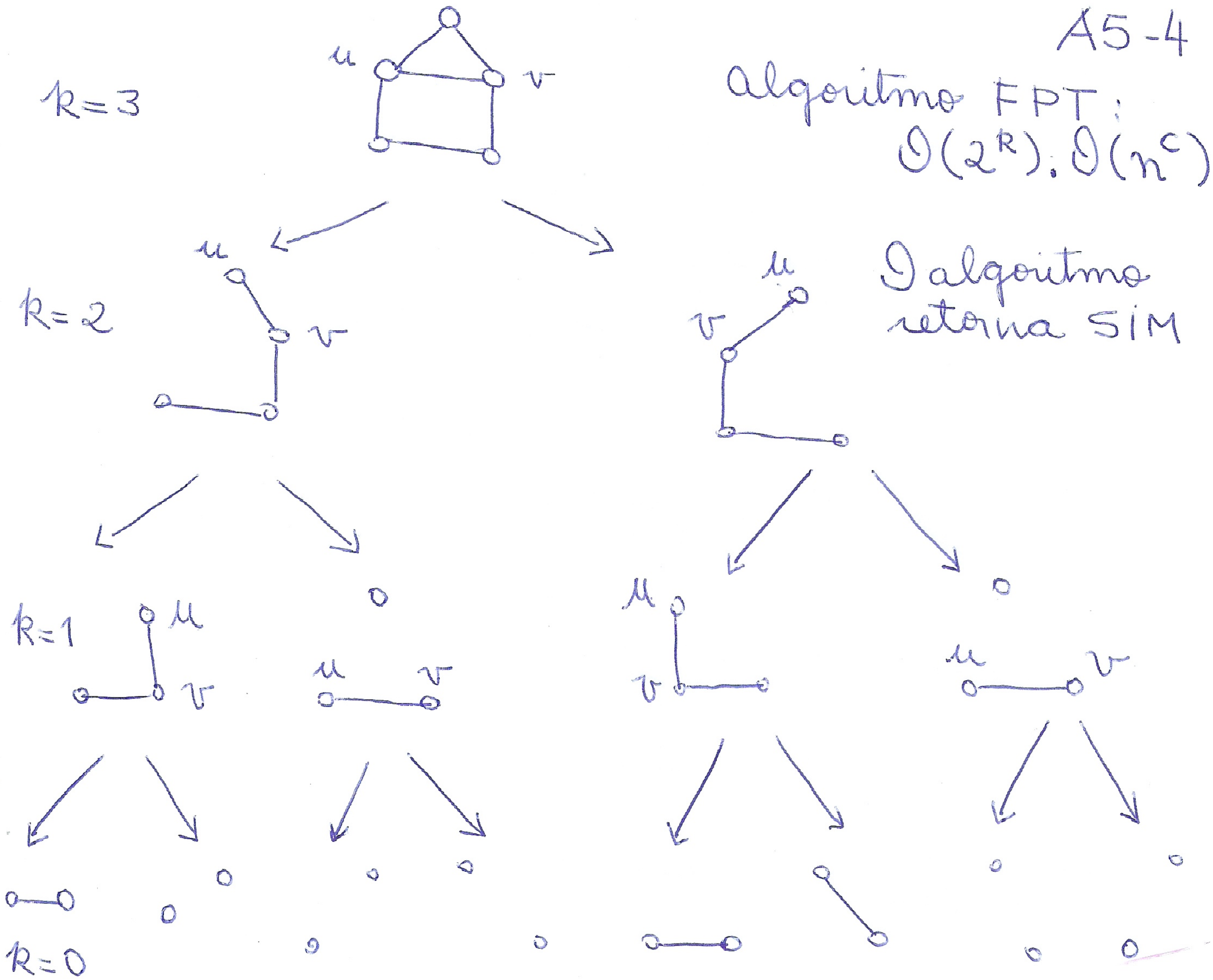
```
CoberturaDeVertices (G, k)
Se G não possui arestas então
    retorna SIM
fim
Se k=0 então
    retorna NÃO
fim
escolha uma aresta uv arbitraria
Se CoberturaDeVertices (G-{u}, k-1) retornar SIM então
    retorna SIM
fim
Se CoberturaDeVertices (G-{v}, k-1) retornar SIM então
    retorna SIM
fim
retorna NÃO
```

Por definição do problema, cada aresta deve possuir pelo menos uma das suas extremidades na cobertura.

A5-4

Algoritmo FPT: $O(2^k) \cdot O(n^c)$

O algoritmo retorna SIM

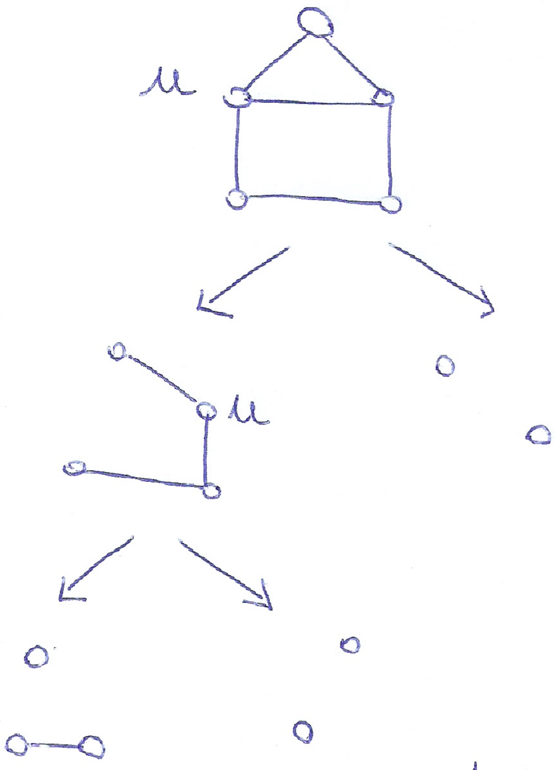


No pior caso, o algoritmo realiza duas chamadas recursivas, tantas vezes quanto possível. Todo nó que potencialmente possa ter 2 filhos fará 2 chamadas recursival. O parâmetro k diminui de 1 a cada nível, o número de folhas é 2^k.

Algoritmo 2

```
CoberturaDeVertices (G,R)
se Δ(G) < 2 então
        se |E(G)| ≤ R entao
                retorna SIM
        senão
                retorna NÃO
        fim
fim
se R = 0 então
        retorna NÃO
fim
seja v um vértice de grau Δ(G)
se CoberturaDeVertices (G-{v}, R-1) retorna SIM então
        retorna SIM
fim
se Δ(G) ≤ R então
        se CoberturaDeVertices(G-{N(v)}, R-Δ(G)) retorna SIM então
                retorna SIM
        fim
fim
retorna NÃO
```

Se $\Delta(G) = 1$, então a cobertura tem um extremo de cada aresta de G.
Se $\Delta(G) \geq 2$, então seja v um vértice de grau máximo. Se v não está na cobertura, então todos os vizinhos de v estão na cobertura.



O algoritmo 1 remove a cada chamada recursiva um vértice, enquanto o algoritmo 2 pode remover um vértice ou todos os seus vizinhos.

Podemos provar que o número máximo de folhas é reduzido neste caso.

Seja $T(n)$ o número máximo de folhas de uma chamada quando o parâmetro é n, portanto $T(k) = T(k-1) + T(k-d(v))$, e como assumimos $d(v) \geq 2$, temos $T(k) \leq T(k-1) + T(k-2)$ e $T(k) = 1$ se $k \leq 1$.

Teorema: Cobertura por Vértices pode ser resolvida em tempo $O(n^{O(1)} \cdot (1{,}6181)^k)$.

prova: uma chamada recursiva tem tempo polinomial $O(n^c)$ para alguma constante. Usaremos a relação de recorrência acima para estabelecer que $T(k) \leq (1{,}6181)^k$. Vale para $k = 0$ e $k = 1$.

$$T(k) \leq T(k-1) + T(k-2) \leq (1{,}6181)^{k-1} + (1{,}6181)^{k-2} \leq (1{,}6181)^{k-2}(1 + 1{,}6181) \leq (1{,}6181)^{k-2} \cdot (1{,}6181)^2 \;\blacksquare$$

Para um Algoritmo 3, basta observar que Cobertura por vértices pode ser resolvido em tempo polinomial se $\Delta(G) \leq 2$.

Modificamos apenas as linhas 2 e 3 do Algoritmo 2:

2. se $\Delta(G) \leqslant 2$ então
3. se existe cobertura de tamanho $\leqslant k$ então

Teorema: Cobertura por Vértices pode ser resolvida em tempo $\Theta(n^{\Theta(1)} \cdot (1{,}4656)^k)$.

prova: Agora no algoritmo 3, as chamadas recursivas só ocorrerão para vértices v, com $d(v) \geqslant 3$ e a relação de recorrência é:

$T(k) \leqslant T(k-1) + T(k-3)$ para $k \geqslant 3$ e
$T(k) = 1$ para $k \leqslant 2$.

Usando que $(1 + 1{,}4656) \leqslant (1{,}4656)^2$ obtemos o resultado. ▨

Para ver o efeito das sucessivas reduções na base da exponencial, verifique a redução no número de nós para $k = 20$, $k = 30$ e $k = 40$.