Complexidade Parametrizada

Matheus Souza D'Andrea Alves

2018.2

Contents

1 Introdução

1.1 Complexidade clássica

Suponha um problema Π, se Π possui um algoritmo que o resolve em tempo polinomial dizemos que $\Pi \in P$.

Se dado um certificado de resposta sim para Π se posso validar tal resposta em tempo polinomial então $\Pi \in NP$.

Chamamos de $NP-Completo$ a classe de problemas Π' no qual dado o problema $3-SAT\ 3S\ 3s \propto \Pi'$.

1.2 Complexidade Parametrizada

1.2.1 O que traz de diferente?

Une teoria e prática, não ignorando nuances práticas de um problema sabidamente NP-Completo, resolve problemas. Difere de heurísticas e aproximações, pois não perde garantia de tempo ou otimalidade.

O objetivo é desenvolver um algoritmo $\mathcal{O}(f(k).n^{\mathcal{O}(1)})$. Se um problema Π admite uma solução dessa forma, dizemos que $\Pi \in FPT$. Quando isso ocorre, afirmamos que existe um pré-processesamento capaz de reduzir a entrada obtendo uma instância menor, limitado por k, chamamos isso de kernelização; Uma solução para o kernel é uma solução para o problema.

Características - $FPT \subset XP$

2 Técnicas de FPT

2.1 Bound search Tree

2.2 Kernelização

Dizemos que se um problema possui um kernel, então por definição o mesmo é FPT.

2.2.1 Pré processamento

chamamos de pré processamento um conjunto de regras que aplicados a uma instancia do problema, pode ou não retirar composições da entrada de forma a podar a instãncia.

Suponha um problema Π, que tem entrada I um parametro k e uma questão q.

Uma kernelização é um algoritmo de pré processamento que recebe I como entrada com o paramêtro k, e retorna uma instância do problema e um inteiro representatne do paramêtro.

$$f(k) \mapsto |I'|$$
$$g(k) \mapsto k'$$

2.2.2 O quão bom pode ser?

Depende de como se planeja abordar o problema, suponha dois algoritmos de kernelização, A e B; Se A chega a um kernel de tamanho $\theta(k)$ em tempo $\theta(n^2)$, e B chega a um tamanho $\theta(k^2)$ em tempo $\theta(n)$ na literatura, diríamos que A é *melhor* pois nos dá um kernel menor.

Teorema 1: Se $(\Pi, k) \in FPT$ então (Π, k) admite um kernel.

Demonstração.
Como $(\Pi, k) \in FPT$ logo o mesmo pode ser resolvido por um algoritmo A em tempo $f(k)n^c$ para c constante.
Considere portanto a seguinte kernelização.

- Execute os n^{c+1} passos de A.
- O problema foi resolvido?
  - Sim:
  - Não: logo $f(k)n^c > n.c \implies f(k) > n$ logo a entrada já era um kernel.

□

2.3 Crown decomposition

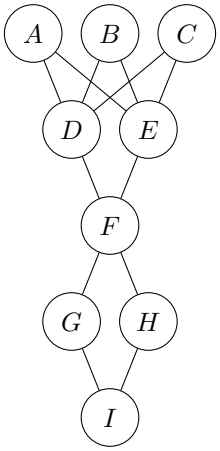

Figure 1:

Estratégia para kernelização via decomposição em coroa

- Encontre uma coroa (C, H, R) de $V(G)$
- Remova $C \cup H$ de G
- $k = k - |H|$

2.3.1 Como encontrar decomposição em coroa em grafos quaisquer

Observe o seguinte lema:

Lema 1: Se G tem mais de $3k$ vértices então G possui
- **um emparelhamento de tamanho $k+1$, ou**
- **uma decomposição em coroa.**

E ambos podem ser encontrados em tempo polinomial.

Demonstração.
Se possui kernel o mesmo possui um emparelhamento com no máximo $2k$ vértices.

$$|V(M)| \leq 2k$$

Observe que isso gera um vertex cover, e portanto $I = G \backslash M$ é independente.

Como queremos encontrar uma cabeça e coroa estamos interessados apenas na vizinhança entre I e M. Ignorando as arestas pertencentes à cabeça, temos um bipartido formado por $I \cup M$. Em I podemos encontrar um novo emparelhamento M' e um conjunto independente I'. Encontrar o vertex cover em bipartidos é polinomial nos dando então um vertex cover VC, é importante notar que em bipartidos o tamanho do emparelhamento máximo é o tamanho do menor vertex cover, logo:

$$|M'| \leq k \quad \text{já que } |M'| \leq |M|/2$$
$$|VC| \leq k$$
$$|M'| = |VC| \leq k$$
$$VC \cap V(M) = \emptyset$$

Seja M^* as arestas de M' que tem um dos vértices em $M \cap VC$.
$H = VC \cap V(M^*)$
$C = V(M^*) \cap I$
$R = V(G) \backslash (C \cup H) V(G) \backslash V(M*)$

□

3 Classes de parametrização

3.1 Valor de entrada

3.2 Subestrutura

3.2.1 Vertex cover

3.2.2 Treewidth

O que é *treewidth*? Devemos primeiro entender decomposição em árvor.

Uma decomposição em árvore de um grafo G é uma estrutura $\mathcal{T} = (T, \chi)$