Parameterised Algorithms

Pradeesha Ashok

International Institute of Information Technology, Bangalore

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

What are parameterised algorithms?

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

NP-Completeness

- Π is NP-Complete

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

NP-Completeness

- Π is NP-Complete
- (Unless $P = NP$) There cannot be an algorithm that solves the general instance of Π exactly, in time polynomial in input size

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

NP-Completeness

- Π is NP-Complete
- (Unless $P = NP$) There cannot be an algorithm that solves the general instance of Π exactly, in time polynomial in input size

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

NP-Completeness

- Π is NP-Complete
- (Unless $P = NP$) There cannot be an algorithm that solves the general instance of Π exactly, in time polynomial in input size

- VERTEX COVER is NP-Complete in general graphs

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

NP-Completeness

- Π is NP-Complete
- (Unless $P = NP$) There cannot be an algorithm that solves the general instance of Π exactly, in time polynomial in input size

- VERTEX COVER is NP-Complete in general graphs
- VERTEX COVER can be solved in polynomial time for trees, bipartite graphs, ...

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

NP-Completeness

- Π is NP-Complete
- (Unless $P = NP$) There cannot be an algorithm that solves the general instance of Π exactly, in time polynomial in input size

- VERTEX COVER is NP-Complete in general graphs

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

NP-Completeness

- Π is NP-Complete
- (Unless $P = NP$) There cannot be an algorithm that solves the general instance of Π exactly, in time polynomial in input size

- VERTEX COVER is NP-Complete in general graphs
- A polynomial time approximation algorithm exists for VERTEX COVER

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Parameterised Algorithms

- The problem instance contains a parameter in addition to the input

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Parameterised Algorithms

- The problem instance contains a parameter in addition to the input
- The value of the parameter is assumed to be much smaller than the input size

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Parameterised Algorithms

- The problem instance contains a parameter in addition to the input
- The value of the parameter is assumed to be much smaller than the input size
- A parameterized algorithm for the problem gives an exact solution in running time which is exponential only in the parameter and polynomial in the input size

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Parameterised Algorithms

Definition

A parameterized problem is a language $L \subset \Sigma \times \mathbb{N}^+$ where Σ is a finite alphabet. For $(x, k) \in \Sigma \times \mathbb{N}^+$, k is called the parameter

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Parameterised Algorithms

Definition

A parameterized problem is a language $L \subset \Sigma \times \mathbb{N}^+$ where Σ is a finite alphabet. For $(x, k) \in \Sigma \times \mathbb{N}^+$, k is called the parameter

Definition

A parameterized problem $L \subset \Sigma \times \mathbb{N}^+$ is called *fixed-parameter tractable* (FPT) if there exists an algorithm $\mathcal{A}$ (called a fixed- parameter algorithm), a computable function f, and a constant c such that, given $(x, k) \in \Sigma \times \mathbb{N}^+$, the algorithm $\mathcal{A}$ correctly decides whether $(x, k) \in L$ in time bounded by $f(k)|(x, k)|^c$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

The Art of Parameterisation

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

The Art of Parameterisation

- Standard Parameter - output size

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

The Art of Parameterisation

- Standard Parameter - output size
- Structural Parameter of the input
  - Graph Problems : max degree, genus, treewidth,....
  - Satisfiablility Problems : Number of clauses, Number of clauses that needs to be satisfied,....
  - Geometric Problems : dimension of the space, number of points,....

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

The Art of Parameterisation

- Standard Parameter - output size
- Structural Parameter of the input
  - Graph Problems : max degree, genus, treewidth,....
  - Satisfiablility Problems : Number of clauses, Number of clauses that needs to be satisfied,....
  - Geometric Problems : dimension of the space, number of points,....
- Parameterization can happen with a combination of two or more values, $k, l, \ldots$ in which case $(k + l + \ldots.)$ is the parameter

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Bounded Search Trees

- Compute a search space - represented by a search tree-of size bounded by a function of the parameter

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER

Definition

GIven a graph $G(V, E)$, $V' \subset V$ is a *vertex cover* for G if for every edge $e \in E$, at least one endpoint of e is in V'

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Vertex Cover

- Input : Undirected graph $G(V, E)$
- Question : Is there a vertex cover V' for G such that $|V'| \leq k$?
- Parameter : k

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for Vertex Cover

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for VERTEX COVER

- For every edge $e(u, v) \in E$ either $u \in V'$ or $v \in V'$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for VERTEX COVER

- For every edge $e(u, v) \in E$ either $u \in V'$ or $v \in V'$ - branch on each endpoint

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for VERTEX COVER

- For every edge $e(u, v) \in E$ either $u \in V'$ or $v \in V'$ - branch on each endpoint
- There exists an FPT algorithm for VERTEX COVER parameterised by k with running time $\mathcal{O}(2^k)$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Improved Branching Algorithm for VERTEX COVER

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Improved Branching Algorithm for VERTEX COVER

- For every vertex $v \in V$, either $v \in V'$ or $N(v) \in V'$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Improved Branching Algorithm for VERTEX COVER

- For every vertex $v \in V$, either $v \in V'$ or $N(v) \in V'$
- Maximum degree is at least 3

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Improved Branching Algorithm for VERTEX COVER

- For every vertex $v \in V$, either $v \in V'$ or $N(v) \in V'$
- Maximum degree is at least 3
- size of the tree $T(k) = T(k-1) + T(k-3)$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Improved Branching Algorithm for VERTEX COVER

- For every vertex $v \in V$, either $v \in V'$ or $N(v) \in V'$
- Maximum degree is at least 3
- size of the tree $T(k) = T(k-1) + T(k-3)$
- There exists an FPT algorithm for VERTEX COVER parameterised by k with running time $\mathcal{O}(1.4656^k)$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Feedback Vertex Set

- Given a graph $G(V, E)$, $F \subset V$ is a *Feedback vertex Set* if $G \setminus F$ is acyclic.

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Feedback Vertex Set

- Input : Undirected graph, $G(V, E)$
- Question : Does G have a feedback vertex set F such that $|F| \leq k$?
- Parameter : k

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for Feedback Vertex Set

- Reduction Rule 1 : If there is a loop at a vertex v, delete v from the graph and decrease k by 1.

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for Feedback Vertex Set

- Reduction Rule 1 : If there is a loop at a vertex v, delete v from the graph and decrease k by 1.
- Reduction Rule 2 : If there is an edge of multiplicity larger than 2, reduce its multiplicity to 2.

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for Feedback Vertex Set

- Reduction Rule 1 : If there is a loop at a vertex v, delete v from the graph and decrease k by 1.
- Reduction Rule 2 : If there is an edge of multiplicity larger than 2, reduce its multiplicity to 2.
- Reduction Rule 3 :If there is a vertex v of degree at most 1, delete v

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for Feedback Vertex Set

- Reduction Rule 1 : If there is a loop at a vertex v, delete v from the graph and decrease k by 1.
- Reduction Rule 2 : If there is an edge of multiplicity larger than 2, reduce its multiplicity to 2.
- Reduction Rule 3 :If there is a vertex v of degree at most 1, delete v
- Reduction Rule 4 : If there is a vertex v of degree 2, delete v and connect its two neighbors by a new edge.

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for Feedback Vertex Set

After applying the reduction rules exhaustively, we get a graph G' that

- contains no loops
- has only single and double edges, and
- has minimum vertex degree at least 3.

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for Feedback Vertex Set

- Let $v_1, v_2, ..., v_n$ be an ordering of the vertices in the descending order of degree
- Let $V_{3k} = \{v_1, v_2, \cdots, v_{3k}\}$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for Feedback Vertex Set

- Let $v_1, v_2, ..., v_n$ be an ordering of the vertices in the descending order of degree
- Let $V_{3k} = \{v_1, v_2, \cdots, v_{3k}\}$

Lemma

Every feedback vertex set in G of size at most k contains at least one vertex of V_{3k}.

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for Feedback Vertex Set

- Let $v_1, v_2, ..., v_n$ be an ordering of the vertices in the descending order of degree
- Let $V_{3k} = \{v_1, v_2, \cdots, v_{3k}\}$

Lemma

Every feedback vertex set in G of size at most k contains at least one vertex of V_{3k}.

- Branch on the choice of vertex from V_{3K}

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Branching Algorithm for FEEDBACK VERTEX SET

- Let $v_1, v_2, ..., v_n$ be an ordering of the vertices in the descending order of degree
- Let $V_{3k} = \{v_1, v_2, \cdots, v_{3k}\}$

Lemma

Every feedback vertex set in G of size at most k contains at least one vertex of V_{3k}.

- Branch on the choice of vertex from V_{3K}
- There exists an FPT algorithm that solves FEEDBACK VERTEX SETin $\mathcal{O}((3k)^k)$ time

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernelization

Preprocess / Reduce the instance I to get an equivalent instance I' such that the size of I' is bounded by a function of the parameter

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernelization

Definition

A *Kernelization algorithm*, or a *kernel*, is an algorithm that given an instance of a parameterised problem (x, k), reduces it to an equivalent instance (x', k') such that $|x'| + k' \leq g(k)$ where g is a computable function.

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for VERTEX COVER

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for VERTEX COVER

- Reduction Rule 1 : If G contains an isolated vertex v, delete v from G

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for VERTEX COVER

- Reduction Rule 1 : If G contains an isolated vertex v, delete v from G
- Reduction Rule 2 : If there is a vertex v of degree at least k + 1, then delete v (and its incident edges) from G and decrement the parameter k by 1

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for VERTEX COVER

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for VERTEX COVER

- If a graph has max degree d, then a set of k vertices can cover at most kd edges

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for VERTEX COVER

- If a graph has max degree d, then a set of k vertices can cover at most kd edges

Lemma

If (G, k) is a yes-instance and none of the reduction rules is applicable to G, then $|V| \leq k^2 + k$ *and* $|E| \leq k^2$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for VERTEX COVER

Theorem

VERTEX COVER *parameterised by output size k has a kernel of $\mathcal{O}(k^2)$ vertices and $\mathcal{O}(k^2)$ edges*

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Edge Clique Cover

- GIven a graph G, decide whether the edges of G can be covered using at most k cliques?

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for EDGE CLIQUE COVER

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for EDGE CLIQUE COVER

- Reduction Rule 1: Remove isolated vertices

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for EDGE CLIQUE COVER

- Reduction Rule 1: Remove isolated vertices
- Reduction Rule 2 : If there is an isolated edge uv (a connected component that is just an edge), delete it and decrease k by 1

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for EDGE CLIQUE COVER

- Reduction Rule 1: Remove isolated vertices
- Reduction Rule 2 : If there is an isolated edge uv (a connected component that is just an edge), delete it and decrease k by 1
- Reduction Rule 3 : If there is an edge uv whose endpoints have exactly the same closed neighborhood, that is, N[u] = N[v], then delete v

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for EDGE CLIQUE COVER

Lemma

If (G,k) is a reduced yes-instance, on which none of the presented reduction rules can be applied, then $|V| \leq 2^k$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Kernel for EDGE CLIQUE COVER

Lemma

If (G,k) is a reduced yes-instance, on which none of the presented reduction rules can be applied, then $|V| \leq 2^k$

Theorem

EDGE CLIQUE COVER *admits a vertex kernel of size* $\mathcal{O}(2^k)$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Iterative Compression

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Iterative Compression

- Consider we are looking for a solution of size k

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Iterative Compression

- Consider we are looking for a solution of size k
- Assume we have a solution Z of slightly larger size $k+1, 2k, \cdots$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Iterative Compression

- Consider we are looking for a solution of size k
- Assume we have a solution Z of slightly larger size $k+1, 2k, \cdots$
- Try to compress Z to size k

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Iterative Compression

- Consider we are looking for a solution of size k
- Assume we have a solution Z of slightly larger size $k+1, 2k, \cdots$
- Try to compress Z to size k

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Iterative Compression

- Consider we are looking for a solution of size k
- Assume we have a solution Z of slightly larger size $k+1, 2k, \cdots$
- Try to compress Z to size k
  - branch in all ways an optimal solution X intersects with Z

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

Iterative Compression

- Consider we are looking for a solution of size k
- Assume we have a solution Z of slightly larger size $k+1, 2k, \cdots$
- Try to compress Z to size k
  - branch in all ways an optimal solution X intersects with Z
- If the compression step can be performed in FPT time, then the problem is in FPT

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

- Start with a 2-approximation to optimal vertex cover , Z

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

- Start with a 2-approximation to optimal vertex cover , Z
- If $|Z| > 2k$, say NO

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

- Let X be a solution of size k

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

- Let X be a solution of size k
- Assume $X_Z = X \cap Z$ (Branch on all possible values of X_Z)

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

- Let X be a solution of size k
- Assume $X_Z = X \cap Z$ (Branch on all possible values of X_Z)
- Let $W = Z \setminus X_Z$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

- Let X be a solution of size k
- Assume $X_Z = X \cap Z$ (Branch on all possible values of X_Z)
- Let $W = Z \setminus X_Z$
- There cannot be an edge in $G[W]$

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

- Let X be a solution of size k
- Assume $X_Z = X \cap Z$ (Branch on all possible values of X_Z)
- Let $W = Z \setminus X_Z$
- There cannot be an edge in $G[W]$
- If $|X_Z \cup N(W)| \leq k$ then say YES

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

- Let X be a solution of size k
- Assume $X_Z = X \cap Z$ (Branch on all possible values of X_Z)
- Let $W = Z \setminus X_Z$
- There cannot be an edge in $G[W]$
- If $|X_Z \cup N(W)| \leq k$ then say YES
- The algorithm runs in $\mathcal{O}(4^k)$ time

Parameterised Algorithms

Pradeesha Ashok

Introduction

Bounded Search Trees

Kernelization

Iterative Compression

VERTEX COVER using Iterated Compression

- Let X be a solution of size k
- Assume $X_Z = X \cap Z$ (Branch on all possible values of X_Z)
- Let $W = Z \setminus X_Z$
- There cannot be an edge in $G[W]$
- If $|X_Z \cup N(W)| \leq k$ then say YES
- The algorithm runs in $\mathcal{O}(4^k)$ time - can be improved to $\mathcal{O}(2^k)$