Parameterized Reoptimization

Felix Reidl

Lehr- und Forschungsgebiet Theoretische Informatik
RWTH Aachen University

August 17, 2010

Contents

The premise for this talk

- Theoretical scenario: solve problem, done.
- Real-life scenario: problem must be solved again and again
- *Small updates*: instance changes only slightly
- *Do we have to throw previous solutions away?*

The premise for this talk

- Theoretical scenario: solve problem, done.
- Real-life scenario: problem must be solved again and again
- *Small updates*: instance changes only slightly
- *Do we have to throw previous solutions away?*

Definition (Reoptimization Scenario)

Input: Instance $\mathcal{I}$, optimal solution S for $\mathcal{I}$, modified instance $\mathcal{I}'$
Problem: Find an optimal solution S' for $\mathcal{I}'$

Note: Difference between $\mathcal{I}'$ and $\mathcal{I}$ is “small”

Previous work in this direction

- So far only tackled with approximation algorithms
- No PTAS $\Rightarrow$ solution quality deteriorates

Previous work in this direction

- So far only tackled with approximation algorithms
- No PTAS $\Rightarrow$ solution quality deteriorates
- PTAS known for VERTEX COVER, INDEPENDENT SET, DOMINATING SET [Bilò,Widmayer,Zych] ...

Previous work in this direction

- So far only tackled with approximation algorithms
- No PTAS $\Rightarrow$ solution quality deteriorates
- PTAS known for Vertex Cover, Independent Set, Dominating Set [Bilò,Widmayer,Zych] ...
- ...but rely on *optimal* solution!

Previous work in this direction

- So far only tackled with approximation algorithms
- No PTAS $\Rightarrow$ solution quality deteriorates
- PTAS known for Vertex Cover, Independent Set, Dominating Set [Bilò,Widmayer,Zych] ...
- ...but rely on *optimal* solution!

$\Rightarrow$ Solving exactly might be a better idea

Parameterized Complexity

- Motivation: more fine-grained look at NP-hard problems
- Isolate "difficult part" of the problem

Parameterized Complexity

- Motivation: more fine-grained look at NP-hard problems
- Isolate “difficult part” of the problem

Definition (Parameterized language)

A *parameterized language* L over an alphabet Σ is a subset of $\Sigma^* \times \mathbb{N}$. For any tuple $(I, k) \in \Sigma^* \times \mathbb{N}$ we call k the *parameter*.

Parameterized Complexity

- Motivation: more fine-grained look at NP-hard problems
- Isolate "difficult part" of the problem

Definition (Parameterized language)

A *parameterized language* L over an alphabet Σ is a subset of $\Sigma^* \times \mathbb{N}$. For any tuple $(I, k) \in \Sigma^* \times \mathbb{N}$ we call k the *parameter*.

Definition (Fixed parameter tractable)

A parameterized language L is contained in the class FPT if there exists an algorithm that decides L in time $f(k)n^{\mathcal{O}(1)}$ for an arbitrary function f.

The famous positive example:

Definition (k-Vertex Cover)

Input: Graph G, integer k
Parameter: k
Problem: Has G a vertex cover of at most size k?

The famous positive example:

Definition (k-Vertex Cover)

Input: Graph G, integer k
Parameter: k
Problem: Has G a vertex cover of at most size k?

- Bounded search tree: pick any uncovered edge (u, v), include either u or v in vertex cover
- After at most k "decisions" we are done
- Search tree has at most 2^k leafs, therefore algorithm runs in $\mathcal{O}(2^k n^{\mathcal{O}(1)})$

The famous negative examples:

- $FPT \subseteq W[1] \subseteq W[2] \ldots$, most likely $FPT \neq W[1]$
- k-INDEPENDENT SET and k-CLIQUE $W[1]$-hard, k-DOMINATING SET $W[2]$-hard

Definition (FPT-Reoptimizable)

A problem L is *FPT-Reoptimizable* (under a specified modification) if the corresponding reoptimization problem can be solved in FPT-time.

Definition (FPT-Reoptimizable)

A problem L is *FPT-Reoptimizable* (under a specified modification) if the corresponding reoptimization problem can be solved in FPT-time.

Let's look at VERTEX COVER again.

Definition (param. VERTEX COVER reoptimization)

Input: Graph G with optimal vertex cover S
Modification: Delete edge e from G to obtain G'
Parameter: $k := |S|$
Problem: Find an optimal vertex cover S' for G'

Definition (FPT-Reoptimizable)

A problem L is *FPT-Reoptimizable* (under a specified modification) if the corresponding reoptimization problem can be solved in FPT-time.

Let's look at VERTEX COVER again.

Definition (param. VERTEX COVER reoptimization)

Input: Graph G with optimal vertex cover S
Modification: Delete edge e from G to obtain G'
Parameter: $k := |S|$
Problem: Find an optimal vertex cover S' for G'

Trivially reoptmizable: use parameterized algorithm for k-VERTEX COVER to solve.

Definition (FPT-Reoptimizable)

A problem L is *FPT-Reoptimizable* (under a specified modification) if the corresponding reoptimization problem can be solved in FPT-time.

Let's look at Vertex Cover again.

Definition (param. Vertex Cover reoptimization)

Input: Graph G with optimal vertex cover S
Modification: Delete edge e from G to obtain G'
Parameter: $k := |S|$
Problem: Find an optimal vertex cover S' for G'

Trivially reoptmizable: use parameterized algorithm for k-Vertex Cover to solve.

Can we do better?

Can we do better?

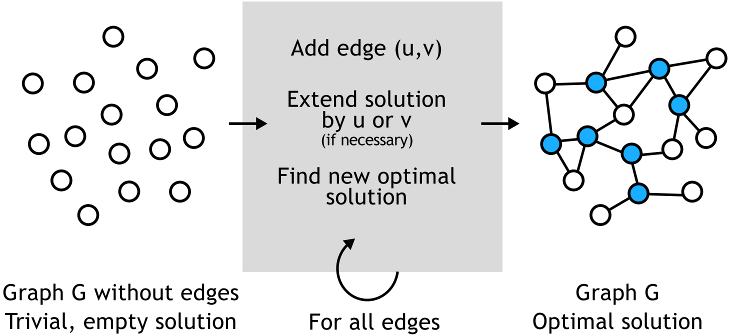
No

Can we do better?

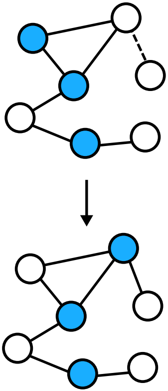
No

(Not by much.)

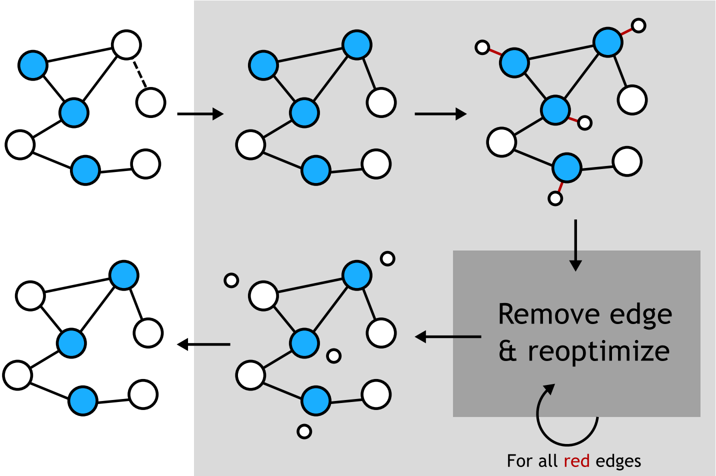
An iterative algorithm



The “compression routine”



The "compression routine"



General result

For all VERTEX COVER-like problems the following holds:

Theorem

Given a reoptimization algorithm $\mathcal{A}$ with running time $f(k)n^{\mathcal{O}(1)}$, one can solve the underlying parameterized problem in time $\mathcal{O}\big(f(k)n^{\mathcal{O}(1)}\big)$.

General result

For all VERTEX COVER-like problems the following holds:

Theorem

Given a reoptimization algorithm $\mathcal{A}$ with running time $f(k)n^{\mathcal{O}(1)}$, one can solve the underlying parameterized problem in time $\mathcal{O}\big(f(k)n^{\mathcal{O}(1)}\big)$.

"Having an optimal solution for a similar instance does not help"

General result

For all Vertex Cover-like problems the following holds:

Theorem

Given a reoptimization algorithm $\mathcal{A}$ with running time $f(k)n^{\mathcal{O}(1)}$, one can solve the underlying parameterized problem in time $\mathcal{O}\big(f(k)n^{\mathcal{O}(1)}\big)$.

"Having an optimal solution for a similar instance does not help"

It does not matter what operation (edge/vertex deletion/addition) the reoptimization algorithm supports.

Affected problems

The theorem holds for the following problems:

- Vertex Cover
- Feedback Vertex Set
- Vertex-Deletion Bipartization
- Cluster Vertex Deletion

Affected problems

The theorem holds for the following problems:

- Vertex Cover
- Feedback Vertex Set
- Vertex-Deletion Bipartization
- Cluster Vertex Deletion

...

- Connected [Insert from above]

Affected problems

The theorem holds for the following problems:

- Vertex Cover
- Feedback Vertex Set
- Vertex-Deletion Bipartization
- Cluster Vertex Deletion

...

- Connected [Insert from above]

Other problems (same result with slightly different approach):

- Steiner Tree

Affected problems

The theorem holds for the following problems:

- Vertex Cover
- Feedback Vertex Set
- Vertex-Deletion Bipartization
- Cluster Vertex Deletion

...

- Connected [Insert from above]

Other problems (same result with slightly different approach):

- Steiner Tree
- Dominating Set (Therefore *not* FPT-Reoptimizable)

Thanks!

Questions?