Decremental Optimization of Dominating Sets Under the Reconfiguration Framework

Alexandre Blanché\textsuperscript{1} \quad Haruka Mizuta\textsuperscript{2} \quad Paul Ouvrard\textsuperscript{1} \\
Akira Suzuki\textsuperscript{2}

\textsuperscript{1}LaBRI, Bordeaux \quad \textsuperscript{2}Tohoku University, Sendai, Japan

June 10, 2020
Dominating sets

Definition

A dominating set is a subset of vertices whose neighborhood contains all the vertices.

Dominating set problem

- **Instance**: A graph $G$, an integer $s$
- **Question**: Does $G$ have a dominating set of size at most $s$?

This problem is NP-complete.
Model: Successive additions and removals of vertices
Model: Successive additions and removals of vertices
Model: Successive additions and removals of vertices
Without a bound on the size of the dominating sets, they are all reachable through successive additions and removals:
We restrict the size of the authorized dominating sets with a bound $k$: 

$|D| \leq k = 4$
Optimization problem

**OPT-DSR** (OPTimization variant of Dominating Set Reconfiguration)

- **Input**: A graph $G$, two integers $k, s$, a dominating set $D_0$ of size $|D_0| \leq k$.
- **Question**: Does $G$ have a dominating set $D_s$ of size $|D_s| \leq s$, such that $D_0 \leftrightarrow^k D_s$?
General complexity

Observation

OPT-DSR generalizes the dominating set problem.

A graph $G = (V, E)$ has a dominating set of size $\leq s$ if and only if the instance $(G, k = |V|, s, D = V)$ is a solution of OPT-DSR.

Alexander Blanché, Haruka Mizuta, Paul Ouvrard, Akira Suzuki  
Decremental Optimization of the Dominating Set Reconfiguration
General complexity

Observation

OPT-DSR generalizes the dominating set problem.

A graph $G = (V, E)$ has a dominating set of size $\leq s$ if and only if the instance $(G, k = |V|, s, D = V)$ is a solution of OPT-DSR.

Corollary

OPT-DSR is NP-hard.
(NP ⊆ PSPACE ⊆ EXPTIME)

Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]

OPT-DSR is PSPACE-complete, including when the input graph:

- is bipartite.

![Graph diagram]
General complexity

\[ \text{(NP} \subseteq \text{PSPACE} \subseteq \text{EXPTIME)} \]

| Theorem [B., Mizuta, Ouvrard, Suzuki (2020)] |
|-----------------------------------------------|
| OPT-DSR is PSPACE-complete, including when the input graph: |
| - is bipartite; |
| - is a split graph; |

![Graphs]
General complexity

\[(\text{NP} \subseteq \text{PSPACE} \subseteq \text{EXPTIME})\]

**Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]**

OPT-DSR is PSPACE-complete, including when the input graph:

- is bipartite;
- is a split graph;
- has bounded *pathwidth*.

Proof: By adapting a result on independent sets and the OPT-ISR problem, analogous to OPT-DSR.
General complexity

Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]

OPT-DSR is PSPACE-complete, including when the input graph:
- is bipartite;
- is a split graph;
- has bounded pathwidth.

Proof: By adapting a result on independent sets and the OPT-ISR problem, analogous to OPT-DSR.
OPT-ISR deals with *independent set* reconfiguration.
**Input**: A graph $G$, $k, s \in \mathbb{N}$, an independent set $I_0$ with $|I_0| \geq k$.

**Question**: Does $G$ have an independent set $I_s$ with $|I_s| \geq s$ and $I_0 \leftrightarrow I_s$?
Theorem [Ito, Mizuta, Nishimura, Suzuki (2018)]
OPT-ISR is PSPACE-hard when the input graph has bounded pathwidth.
Idea of the reduction

An independent set

A vertex cover (its complement)

An equivalent dominating set

Corollary

**OPT-ISR** is PSPACE-hard $\Rightarrow$ **OPT-VCR** is PSPACE-hard

Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]

**OPT-DSR** is PSPACE-hard.
Positive results

Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]

OPT-DSR can be solved in polynomial time on interval graphs.
Proof : \((G, k, s, D)\), with \(G\) an interval graph
Interval graphs

Proof: 

\((G, k, s, D)\), with \(G\) an interval graph

- We build in linear time in \(|G|\) a possible representation of \(G\) as a set of intervals.
**Interval graphs**

*Proof*: \((G, k, s, D)\), with \(G\) an interval graph

- We build a **minimum dominating set** \(D_m\) of \(G\), in linear time in \(|G|\).

**Lemma [Haddadan et al. (2015)]**

We can reconfigure \(D\) in \(D'\), s.t. \(D_m \subseteq D'\), under the bound \(|D| + 1\), in linear time in \(|G|\).
Proof: \((G, k, s, D)\), with \(G\) an interval graph

We build a minimum dominating set \(D_m\) of \(G\), in linear time in \(|G|\).

We thus have \(D \leftrightarrow^k D' \leftrightarrow^k D_m\).

We can answer yes if \(|D_m| \leq s\) and produce the corresponding sequence in linear time in \(|G|\).
Interval graphs

Proof: We build a minimum dominating set of $G$.

- Ordering: by ending time
- favored neighbor of $v_i$ := maximum neighbor of $v_i$ in the ordering

Algorithm
Traverse the vertices in order.
If $v_i$ is dominated, skip it.
Otherwise, add its favored neighbor to the dominating set.
**Proof**: We build a minimum dominating set of $G$.

- **Ordering**: by ending time
- **favored neighbor** of $v_i :=$ maximum neighbor of $v_i$ in the ordering

**Algorithm** Traverse the vertices in order.

- If $v_i$ is dominated, skip it.
- Otherwise, add its favored neighbor to the dominating set.
Interval graphs

**Proof**: We build a minimum dominating set of $G$.

- **Ordering**: by ending time
- **favored neighbor** of $v_i :=$ maximum neighbor of $v_i$ in the ordering

**Algorithm** Traverse the vertices in order.

- If $v_i$ is dominated, skip it.
- Otherwise, add its favored neighbor to the dominating set.
Proof: We build a minimum dominating set of $G$.

- Ordering: by ending time
- favored neighbor of $v_i :=$ maximum neighbor of $v_i$ in the ordering

Algorithm Traverse the vertices in order.
- If $v_i$ is dominated, skip it.
- Otherwise, add its favored neighbor to the dominating set.
Interval graphs

**Proof**: We build a minimum dominating set of $G$.

- **Ordering**: by ending time
- **Favored neighbor** of $v_i$ := maximum neighbor of $v_i$ in the ordering

**Algorithm** Traverse the vertices in order.
- If $v_i$ is dominated, skip it.
- Otherwise, add its favored neighbor to the dominating set.
Interval graphs

**Proof**: We build a minimum dominating set of $G$.

- **Ordering**: by ending time
- **favored neighbor** of $v_i :=$ maximum neighbor of $v_i$ in the ordering

**Algorithm** Traverse the vertices in order.

- If $v_i$ is dominated, skip it.
- Otherwise, add its favored neighbor to the dominating set.
Interval graphs

Proof: We build a minimum dominating set of $G$.

- **Ordering**: by ending time

- *favored neighbor* of $v_i :=$ maximum neighbor of $v_i$ in the ordering

### Algorithm

Traverse the vertices in order.

- If $v_i$ is dominated, skip it.
- Otherwise, add its favored neighbor to the dominating set.
**Proof**: We build a minimum dominating set of $G$.

- **Ordering**: by ending time

- *favored neighbor* of $v_i$ := maximum neighbor of $v_i$ in the ordering

**Algorithm** Traverse the vertices in order.

- If $v_i$ is dominated, skip it.
- Otherwise, add its favored neighbor to the dominating set.
Proof: We build a minimum dominating set of $G$.

- Ordering: by ending time

- favored neighbor of $v_i :=$ maximum neighbor of $v_i$ in the ordering

Algorithm
Traverse the vertices in order.

- If $v_i$ is dominated, skip it.
- Otherwise, add its favored neighbor to the dominating set.
Proof: We build a minimum dominating set of $G$.

- Ordering: by ending time
- *favored neighbor* of $v_i :=$ maximum neighbor of $v_i$ in the ordering

**Algorithm** Traverse the vertices in order.

- If $v_i$ is dominated, skip it.
- Otherwise, add its favored neighbor to the dominating set.
A graph $G$ is $d$-degenerate if it possesses a vertex $v$ of degree $\leq d$ and $G - v$ is also $d$-degenerate.

A 2-degenerate graph.

**Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]**

OPT-DSR can be solved in time $\text{FPT}(d + s)$, i.e. in time $f(d + s) \times n^{O(1)}$ if $|G| = n$; where $d$ is the degeneracy of the graph and $s$ is the size of the sought solution.
A graph $G$ is $d$-degenerate if it possesses a vertex $v$ of degree $\leq d$ and $G - v$ is also $d$-degenerate.

A 2-degenerate graph.

**Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]**

OPT-DSR can be solved in time $\text{FPT}(d + s)$, i.e. in time $f(d + s) \times n^{O(1)}$ if $|G| = n$; where $d$ is the degeneracy of the graph and $s$ is the size of the sought solution.
A graph $G$ is $d$-degenerate if it possesses a vertex $v$ of degree $\leq d$ and $G - v$ is also $d$-degenerate.

![A 2-degenerate graph.](image)

**Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]**

OPT-DSR can be solved in time $\text{FPT}(d + s)$, i.e. in time $f(d + s) \times n^{O(1)}$ if $|G| = n$; where $d$ is the degeneracy of the graph and $s$ is the size of the sought solution.
A graph $G$ is $d$-degenerate if it possesses a vertex $v$ of degree $\leq d$ and $G - v$ is also $d$-degenerate.

A 2-degenerate graph.

**Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]**

OPT-DSR can be solved in time $\text{FPT}(d + s)$, i.e. in time $f(d + s) \times n^{O(1)}$ if $|G| = n$; where $d$ is the degeneracy of the graph and $s$ is the size of the sought solution.
A graph $G$ is $d$-degenerate if it possesses a vertex $v$ of degree $\leq d$ and $G - v$ is also $d$-degenerate.

A 2-degenerate graph.

**Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]**

OPT-DSR can be solved in time $\text{FPT}(d + s)$, i.e. in time $f(d + s) \times n^{O(1)}$ if $|G| = n$; where $d$ is the degeneracy of the graph and $s$ is the size of the sought solution.
Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]

OPT-DSR can be solved in time $\text{FPT}(\tau)$, where $\tau$ is the size of a minimum vertex cover of the graph.
Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]

OPT-DSR can be solved in time $\text{FPT}(\tau)$, where $\tau$ is the size of a minimum vertex cover of the graph.

Proof:

- **Trivial case**: If $|D| = k$ and $D$ is minimal, then $D$ is frozen. We cannot remove nor add vertices, hence the instance $(G, k, s, D)$ is positive iff $|D| \leq s$. → We can test this condition in time $O(|G|)$.

Exemple pour $k = 3$
Parameterized complexity

Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]

OPT-DSR can be solved in time \( \text{FPT}(\tau) \), where \( \tau \) is the size of a minimum vertex cover of the graph.

Proof:

- **Trivial case**: If \( |D| = k \) and \( D \) is minimal, then \( D \) is frozen. We cannot remove nor add vertices, hence the instance \((G, k, s, D)\) is positive iff \( |D| \leq s \).
  \( \rightarrow \) We can test this condition in time \( O(|G|) \).

- If \( |D| = k \) and \( D \) is not minimal, then we can remove a vertex of \( D \) and reduce to the last case, \( |D| < k \).

From now on we assume that \( |D| < k \).

\( \rightarrow \) We can add at least 1 vertex to \( D \) without getting above \( k \).
Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]

OPT-DSR can be solved in time FPT(τ), where τ is the size of a minimum vertex cover of the graph.

- We compute τ in time FPT(τ).
- 2 possibilities:
  - Either τ > s: As d ≤ τ, we have d + s ≤ 2τ.
    → we use the FPT(d + s)-time algorithm ✓
Parameterized complexity

**Theorem [B., Mizuta, Ouvrard, Suzuki (2020)]**

OPT-DSR can be solved in time FPT($\tau$), where $\tau$ is the size of a minimum vertex cover of the graph.

- We compute $\tau$ in time FPT($\tau$).
- 2 possibilities:
  - Either $\tau > s$: As $d \leq \tau$, we have $d + s \leq 2\tau$. → we use the FPT($d + s$)-time algorithm ✓
  - Or $\tau \leq s$: The instance is a **positive instance** in this case. To prove it, we will reconfigure $D$ into a dominating set $D'$ that satisfies $|D'| \leq \tau$. 

Alexandre Blanché, Haruka Mizuta, Paul Ouvrard, Akira Suzuki

Decremental Optimization of the Dominating Set Reconfiguration
Proof

Let $G$ be a graph.
We build a minimum vertex cover in time $\text{FPT}(\tau)$.\text{\hspace{1cm}}
Proof

Let $D$ be an initial dominating set.
Proof

We associate to each $v \in X \setminus D$ a neighbor $t(v)$. Let $T = \{t(v) \mid v \in X \setminus D\}$.
Proof

X I

vertex cover

stable

Already dominated (∅ ∈ T)

Alexandre Blanché, Haruka Mizuta, Paul Ouvrard, Akira Suzuki
Decremental Optimization of the Dominating Set Reconfiguration
Proof

vertex cover

stable

X

I
Proof

vertex cover
stable
Already dominated ($\not\in T$)

X

I
Proof

vertex cover

stable

Not dominated ($\notin T$)

X

I
Proof
Proof

X

vertex cover

stable

I
Proof

X

stable

vertex cover

Already dominated ($\emptyset \in T$)

I
Proof

vertex cover

stable

X

I

$\in T$

$\in T$
When \((I \cap D') \subseteq J\), we have:

\[
|I \cap D'| \leq |t^{-1}(I \cap D')| \\
\leq |X \setminus D'| \\
= \tau - |X \cap D'|
\]

Hence we have:

\[
|D'| = |I \cap D'| + |X \cap D'| \leq \tau \leq s
\]

Thus \(D'\) is a solution: the instance is positive.
Proof

When \((I \cap D') \subseteq J\), we have:

\[
|I \cap D'| \leq |t^{-1}(I \cap D')| \\
\leq |X \setminus D'| \\
= \tau - |X \cap D'|
\]

Hence we have:

\[
|D'| = |I \cap D'| + |X \cap D'| \\
\leq \tau \\
\leq s
\]

Thus \(D'\) is a solution:
the instance is positive.
### Complexity of OPT-DSR

- **PSPACE-complete** (even on bipartite, split and bounded pathwidth graphs)
- Polynomial on Interval graphs
- **FPT**\((d + s)\) \((d = \text{degeneracy}, s = \text{size of the solution})\)
- **FPT**\((\tau)\) \((\tau = \text{size of a minimum vertex cover})\)
Complexity of OPT-DSR

- PSPACE-complete (even on bipartite, split and bounded pathwidth graphs)
- Polynomial on Interval graphs
- FPT($d + s$) \((d = \text{degeneracy}, s = \text{size of the solution})\)
- FPT($\tau$) \((\tau = \text{size of a minimum vertex cover})\)

Thanks for your attention.