Parameterized Complexity of Reconfiguration of Atoms

Appeared at 16th International Conference and Workshops on Algorithms and Computation

Alexandre Cooper[1], **Stephanie Maaz**[2,*], Amer E. Mouawad[3], Naomi Nishimura[2,*]

[1]Institute of Quantum Computing, University of Waterloo, Canada

[2]David R. Cheriton School of Computer Science, University of Waterloo, Canada

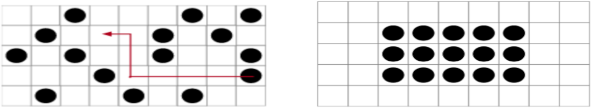
[3]Department of Computer Science, American University of Beirut, Lebanon

[3]University of Bremen, Bremen, Germany

* Research supported by the Natural Sciences and Engineering Research Council of Canada.

Motivation: Challenges in Quantum Simulation

- Given a positioning of a set of traps, loading atoms into those traps results in a random non-desired arrangement of atoms.
- Can move an atom along a connected series of traps that are ***empty***.
- Survival probability of an atom decreases due to movement.
- **Goal:** Minimize the total number of moves.



A randomly generated 2D-positioning of atoms in a 2D-array of traps.

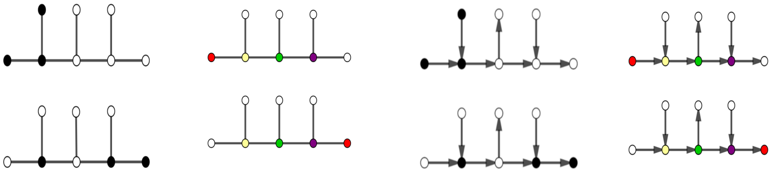
[Schymik et al., 2020]
[Ebadi et al., 2021]

A Reconfiguration Problem

- This problem can be seen as a reconfiguration problem. For a definition of reconfiguration problems, see [Ito et al., 2011].
- **Configuration**: set of vertices representing the placement of tokens in a graph G.
- **Move**: displacement of a single token along a path of free vertices (vertices without tokens).
- **Transforming sequence**: sequence of moves so that we form a target configuration T from a source configuration S of a given graph G.
- $|S| = |T|$.

Token Moving (TM): For a given graph G, source configuration S, and target configuration T, can we find a transforming sequence of length at most ℓ?

Token Moving Is NP-Hard



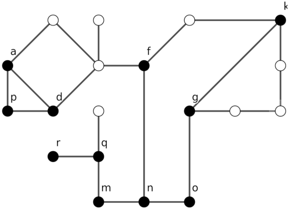
Unlabelled **U**ndirected | **L**abelled **U**ndirected | **U**nlabelled **D**irected | **L**abelled **D**irected

- It is **NP-hard** for both undirected variants [Calinescu et al., 2018].
- **UDTM** and **LDTM** are also **NP-hard**.

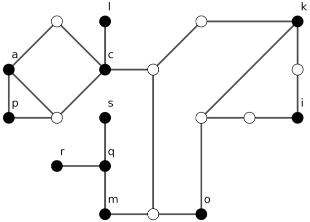
Parameterized Algorithms and Complexity

- Design algorithms to solve problems in time $f(p) \cdot poly(n)$, where:
  - n is the size of the instance,
  - p is some parameter(s).
- Intuition: design algorithms that put all the load on the parameters.
- A problem is **fixed-parameter tractable** if it admits such an algorithm.
- Analogous to P: *FPT*.
  Analogous to NP-hard: $W[1]$-hard or $W[2]$-hard.

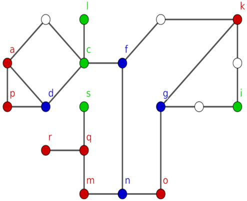
Terminology - UTM



Representation of S on G



Representation of T on G



- **O** (for ***obstacle vertices***): $S \cap T$ (red).
- **T\S** (green).
- **S\T** (blue).
- **F** (for ***free vertices***): V_G - $S \cup T$ (white).

Outline

Possible parameters:

- k, the number of tokens
- ℓ, the number of moves
- f, the number of free vertices

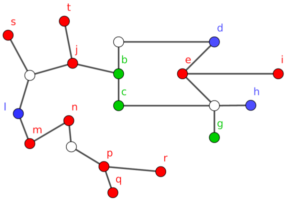
Below are the proven results in the paper:

	k	ℓ	$\ell + f$	$\ell - \lvert S \setminus T \rvert$
UUTM	**FPT**	FPT	FPT	W[2]-hard
UDTM	**FPT**	FPT	FPT	W[2]-hard
LUTM	Open	W[1]-hard	W[1]-hard	W[2]-hard
LDTM	Open	W[1]-hard	W[1]-hard	W[2]-hard

Table: Summary of results for **U**nlabelled/**L**abelled and **U**ndirected/**D**irected **T**oken **M**oving problem variants

Parameter k - UUTM & UDTM

- k: the number of tokens.
- Build an **equivalent** smaller instance, of size some function of k; instance with shortest transforming sequences of the same length to those of the original instance.



- $f = |F|$, where F is the set of free vertices.
- $n - f \leq 2k$.

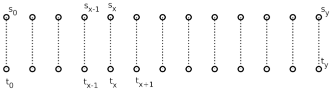
Parameter k - UUTM & UDTM

Lemma 1.1

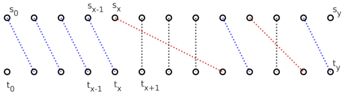
For any yes-instance of **UUTM** or any instance of **UDTM**, in a shortest transforming sequence, no token moves more than once.

Proof by contradiction:

- Pick a shortest sequence that **minimizes** the distance between the first and the second move of the same token ($t_0 = s_y$).
- Build a new sequence with **one less move** and maintain the invariant that the two sequences differ only in the placement of a single token.



β



β'

Parameter k - UUTM & UDTM

Lemma 1.2

For any instance of **UUTM** or any instance of **UDTM**, we can form an **equivalent contracted instance**.

- The only role a free vertex can play is in connecting its neighbors, thus remove it and add an edge (arc) between each appropriate pair of its neighbors.

Lemma 1.3

UUTM and **UDTM** are fixed-parameter tractable and can be solved in time $k^{O(\ell)} \cdot n^{O(1)}$, where k is the number of tokens and ℓ is the number of moves.

- Choose up to 2ℓ vertices from $S \cup T$, pair them as sources and targets of moves, order those moves, and test in polynomial time whether the formed sequence is a transforming sequence.

Road Map

- k, the number of tokens
- ℓ, the number of moves
- f, the number of free vertices

	k	ℓ	$\ell + f$	$\ell - \|S \setminus T\|$
UUTM	FPT	**FPT**	FPT	W[2]-hard
UDTM	FPT	FPT	FPT	W[2]-hard
LUTM	Open	W[1]-hard	W[1]-hard	W[2]-hard
LDTM	Open	W[1]-hard	W[1]-hard	W[2]-hard

Table: Summary of results for **U**nlabelled/**L**abelled and **U**ndirected/**D**irected **T**oken **M**oving problem variants

Parameter ℓ - UUTM

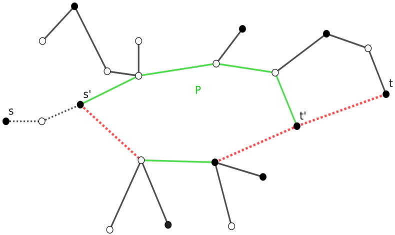
G_α: graph resulting from removing from the representation of the source configuration on G any parts not used by a sequence of moves α.
$\rightarrow$Every token appearing in G_α participates in at least one move.

Lemma 2.1

For any contracted instance of **UUTM**, there exists a transforming sequence α of minimum length such that G_α is a forest.

Proof by contradiction:

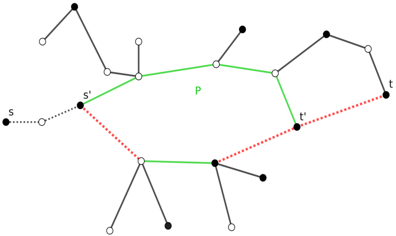
- Pick any sequence and look at the subsequence β between the first move in α and the move that form the first cycle(s) in G_α.
- Each token in G_β must move (once).



Parameter ℓ - UUTM

Proof by contradiction:

- Build from G_β a forest of trees with equal number of vertices in S and T.
- We can find a minimum length sequence for any instance of **UUTM** in linear time on trees [Calinescu et al., 2018].
- Repeat the reasoning for the next cycle(s) in G_α.



Parameter ℓ - UUTM - Proof

Lemma 2.2

For a contracted instance of **UUTM**, there exists a transforming sequence α of minimum length such that G_α is a forest, each tree in the forest is a **minimum Steiner tree** with terminals and leaves in $S \Delta T$, internal vertices in $S \cup T$, and such that each internal vertex in O is the source vertex of a move.

Finding a minimum Steiner tree is fixed-parameter tractable when parameterized by the number of terminals [Dreyfus & Wagner, 1972].

Theorem 2.1

***UUTM** is fixed-parameter tractable when parameterized by ℓ, the number of moves.*

- Form an equivalent contracted instance.
- Attempt all possible partitions of vertices in $S \Delta T$ into $1, \ldots, \ell$ Steiner trees, having equal number of vertices in $S \setminus T$ and $T \setminus S$.
- The number of moves associated with each tree is equal to the number of tokens present in the tree.

Road Map

- k, the number of tokens
- ℓ, the number of moves
- f, the number of free vertices

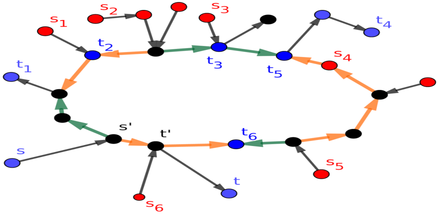
	k	ℓ	$\ell + f$	$\ell - \|S \setminus T\|$
UUTM	FPT	FPT	FPT	W[2]-hard
UDTM	FPT	**FPT**	FPT	W[2]-hard
LUTM	Open	W[1]-hard	W[1]-hard	W[2]-hard
LDTM	Open	W[1]-hard	W[1]-hard	W[2]-hard

Table: Summary of results for **U**nlabelled/**L**abelled and **U**ndirected/**D**irected **T**oken **M**oving problem variants

Parameter ℓ - UDTM

Lemma 3.1

If there exist instances of **UDTM** such that for every transforming sequence α of minimum length, G_α is not a forest, then at least one of those instances must be a **contracted circle instance**:



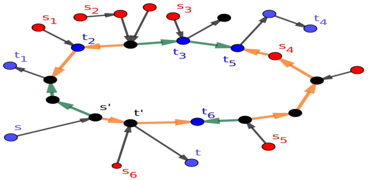
- Cycle vertices and cycle segments.
- Forest of trees attached to the cycle vertices, where in each tree all arcs are directed solely towards or solely away from the root.
- Source (sink) junction vertices with an out-pool (in-pool) tree.

Parameter ℓ - UDTM

Lemma 3.2

Given a directed tree D, two configurations S and T of D such that every leaf of D is in $S \Delta T$, and a **one-to-one mapping** μ from S to T such that there is a directed path from each $s \in S$ to $\mu(s) \in T$ (and $s \neq \mu(s)$ for all s), then there exists a transformation from S to T in D.

- Find a **one-to-one mapping** that does not use $s't'$ in the **contracted circle instance**.



Lemma 3.3

For any yes-instance **UDTM**, there exists a transforming sequence α of minimum length such that G_α is a directed forest.

Parameter ℓ - UDTM

[Alon et al., 2008.]
Let H be a directed forest on q vertices. Let $D = (V, E)$ be a directed n-vertex graph and $\beta : E \to \mathcal{R}$ be a real-weight function defined on the edges of D, then a subgraph of D isomorphic to H with maximal total weight, if one exists, can be found in FPT worst-case time.

Theorem 3.1

UDTM *is fixed-parameter tractable when parameterized by* ℓ.

- Form an equivalent contracted instance of the given graph D.
- q, the total number of vertices in D_α is at least $|S \Delta T|$ and at most $|S \Delta T| + \ell - S \setminus T$.
- Enumerate all directed forests (H) on q vertices, with the sets S', T' and determine in fixed parameter tractable time whether it is a yes-instance.
- Assign weights to edges of the graph D and add edges to D and H so as to use the theorem of Alon et al. to find if D contains a subgraph of the correct form to be isomorphic to D_α.

Road Map

- k, the number of tokens
- ℓ, the number of moves
- f, the number of free vertices

	k	ℓ	$\ell + f$	$\ell - \|S \setminus T\|$
UUTM	FPT	FPT	FPT	**W[2]-hard**
UDTM	FPT	FPT	FPT	**W[2]-hard**
LUTM	Open	W[1]-hard	W[1]-hard	**W[2]-hard**
LDTM	Open	W[1]-hard	W[1]-hard	**W[2]-hard**

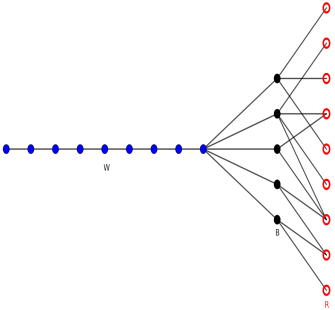
Table: Summary of results for **U**nlabelled/**L**abelled and **U**ndirected/**D**irected **T**oken **M**oving problem variants

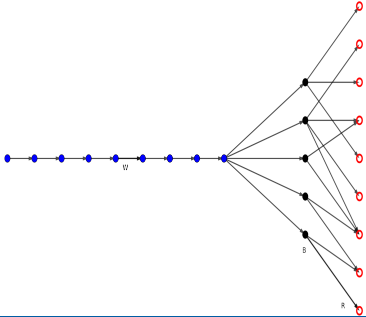
Parameter ℓ - S \T

Red-Blue Dominating Set (RBDS): For a bipartite graph $G = (V_B \cup V_R, E)$ of blue and red vertices and an integer k, determine whether G contains a subset of V_B of size at most k such that each vertex in V_R is the neighbor of a vertex in the subset. RDBS is W[2]-hard. [Downey & Fellows, 1997]

Using Red-Blue Dominating Set, **UUTM** and **UDTM** are W[2]-hard when parameterized by $\ell - |S \backslash T|$.

- $\ell = |R| + k$.

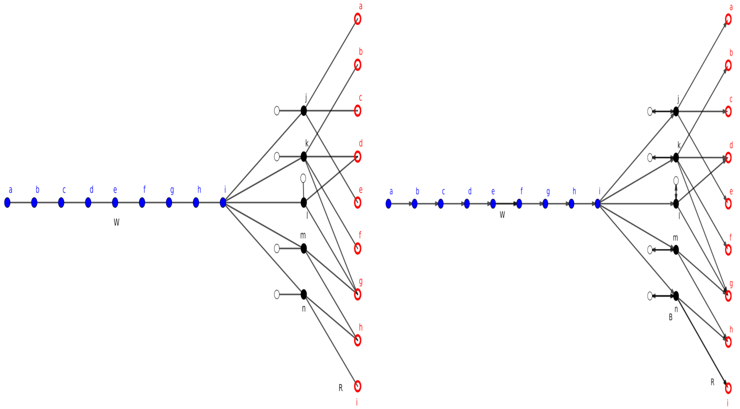




Parameter ℓ - S \T

Using Red-Blue Dominating Set, **LUTM** and **LDTM** are W[2]-hard when parameterized by $\ell - |S \backslash T|$.

- $\ell = |R| + 2k$.



Future Work and Open Questions

Other challenges present in the process:

- Under certain conditions, atoms can be displaced simultaneously.
- Survival probability of an atom decreases also with the distance it travels and the passage of time.

Open questions:

- Can we find efficient approximation algorithms with provable guarantees?
- Can we also design efficient parallel approximation algorithms?
- Can we incorporate movement of atoms in batches subject to a given set of physical constraints?

Thank you!
Any questions?