Diversity of Solutions: An Exploration Through the Lens of Fixed-Parameter Tractability Theory

Julien Baste [1] Michael R. Fellows [2] Lars Jaffke [2] Tomáš Masařík [3,4] Mateus de Oliveira Oliveira [2] Geevarghese Philip [5] Frances A. Rosamond [2]

[1]University of Lille, France [2]University of Bergen, Norway [3]University of Warsaw, Poland [4]Charles University, Czech Republic [5]Chennai Mathematical Institute and UMI ReLaX, India

The Problem

Finding *one* good solution to an algorithmic problem is often not of much use to the end user.

An Example

A *vertex cover* of a graph G is any subset S of its vertices such that *deleting* S from G gives a graph with no edges. The **Vertex Cover** problem asks for a *smallest* vertex cover of an input graph.

Vertex Cover is used to model *conflict resolution*, for instance in helping Air Traffic Controllers (ATCs) prevent aircraft collisions. Each vertex in this *conflict graph* G is an aircraft. There is an edge between two vertices if the two aircraft risk *interference*. A *vertex cover* of G gives a set of aircraft which can be asked to change course to *eliminate* the risk of interference. A *smallest* vertex cover gives a *smallest* set of aircraft to redirect. *Ergo*, a fast algorithm that solves the **Vertex Cover** problem is of great use to ATCs.

...or is it?

Not really!

Not every choice of (say) ten aircraft to redirect is equally desirable.

It is likely better to make

- smaller aircraft to change course, instead of larger ones;
- cruising aircraft to change course, instead of those which are taking off or landing;

...and so on.

An algorithm that finds an arbitrary vertex cover of small size is likely of no help to an ATC.

This Is Ubiquitous

This is an issue with most "neat" algorithmic problems derived from the real world. The modelling process throws away so much *side information* that **one optimal solution** to the final algorithmic problem **is next to useless** to the practitioner.

Finding *all* optimal solutions is infeasible for most problems. And finding many optimal solutions which are similar each other is not much more useful than finding one optimal solution.

Our Solution

We design fast algorithms to find a *small number* of *good quality solutions* which are *dissimilar* to one another. We call this a "diverse" set of solutions.

Given a diverse set of solutions, an end-user can choose one solution by factoring in the *side information* that was lost during the modelling.

The Example, Again

Our algorithm would find a collection of (say) five sets of (say) ten aircraft each such that

- Redirecting all the ten aircraft in any one set removes the risk of interference, and
- No two of these sets have (say) seven or more aircraft in common.

Such a collection of good, diverse solutions would be much more useful to the ATC than just one set of ten aircraft to divert.

The Setting

We study diverse variants of *vertex-problems* on graphs. These are problems where the input is a graph G and we are looking for an optimal (small, or large, as the case may be) subset S of vertices of G which satisfies some desired property. **Vertex Cover** is an example of a vertex-problem, and so are hundreds of other problems of great practical interest such as **Feedback Vertex Set**, **Dominating Set**, and **Independent Set**.

Our **diversity measure** for a collection of solutions is the *sum of pairwise Hamming distances* of the vertex sets which form the solutions in the collection. Our aim is to find a *small* collection of optimal solutions whose sum of pairwise Hamming distances is *large*.

For most interesting vertex-problems—including **Vertex Cover**, **Feedback Vertex Set**, **Dominating Set**, and **Independent Set**—finding *one* optimal solution is already NP-hard, and so is finding a *collection* of such solutions. Hence we look at problems for which finding one optimal solution is *fixed-parameter tractable* (FPT) for a natural parameter, namely the *treewidth* t of the input graph G.

The *treewidth* t of graph G is a measure of how "tree-like" it is, and graphs derived from real-world instances of a surprising variety of problems have been observed to have low treewidth.

An FPT algorithm with treewidth t as the parameter solves the problem in $\mathcal{O}^\star(f(t))$ time, where the $\mathcal{O}^\star()$ notation hides polynomial factors in the size of the input graph G. For instance, there are FPT algorithms that solve **Vertex Cover**, **Feedback Vertex Set**, **Dominating Set**, and **Independent Set** in running times of the form $\mathcal{O}^\star(c^t)$ for (different) small constants c. Note that when real-world input graphs are (empirically) guaranteed to have bounded treewidth t these algorithms are **effectively polynomial-time algorithms for inputs that matter**, though the problems themselves are NP-hard. This is the **great appeal of fixed-parameter tractability**.

All currently known practical FPT algorithms for such parameterizations of vertex-problems are derived using Dynamic Programming (DP) on *tree decompositions* of width t. The typical parameterization of a vertex-problem by treewidth is as follows:

Vertex-Problem

Input:	A graph G, a tree decomposition of G of width t, and a positive integer k.
Parameter:	t
Task:	Find a solution of size k to the vertex-problem for G, or report correctly that no such solution exists.

Diverse Problems

We consider diverse versions of vertex-problems parameterized by both the treewidth and the number of solutions. The typical parameterization is as follows:

Diverse Vertex-Problem

Input:	A graph G, a tree decomposition of G of width t, and positive integers k, r, d.
Parameter:	t, r
Task:	Find a **collection** $\mathcal{S} = \{S_1, S_2, \ldots, S_r\}$ of r solutions each of size k to the vertex-problem for G such that the **diversity** $D(\mathcal{S})$ is *at least* d, or report correctly that no such collection exists.

The *diversity* $D(\mathcal{S})$ is defined as

$$D(\mathcal{S}) = \sum_{i \neq j \in [r]} Ham(S_i, S_j),$$

where $Ham(S_i, S_j) = |S_i \setminus S_j| + |S_j \setminus S_i|$ is the *Hamming distance* of the pair (S_i, S_j).

Our Results

Our main result is that if a vertex-problem is FPT parameterized by treewidth via an algorithm that does DP on tree decompositions, then its diverse variant is also FPT, parameterized by treewidth and the number of solutions.

Theorem

If **Vertex-Problem** can be solved in $\mathcal{O}^\star(f(t))$ time by dynamic programming on tree decompositions then **Diverse Vertex-Problem** can be solved in $\mathcal{O}^\star(f(t)^r)$ time.

Our proof of this theorem in fact shows how to *automatically transform* a such a DP algorithm for any **Vertex-Problem** to an algorithm that solves the corresponding **Diverse Vertex-Problem** within the above running time bounds.

Note that the diversity bound d (which can be as big as $r^2 \cdot n$) does **not** appear in the running time bound. This is because the dependence of the running time on d is **polynomial**. A naïve dynamic programming algorithm for **Diverse Vertex-Problem** would have a running time of the form $\mathcal{O}^\star(d^{\mathcal{O}(r^2)} f(t)^r)$.

The Example, One Last Time

It is known that the treewidth of a graph G cannot be much more than the size of its smallest vertex cover. This allows us to solve **Diverse Vertex Cover** using the above theorem, even if the tree decomposition is **not** part of the input. In the following the integers k, r, d have the same meanings as in the definition of **Diverse Vertex-Problem**.

Theorem

Let G be a graph, and let k, r, d be integers. There is an algorithm which solves the **Diverse Vertex Cover** problem for inputs (G, k, r, d) in time $\mathcal{O}^\star((2^{(k+2)} \cdot (k+1))^r)$.

More Results: Kernelization

The notion of **kernelization** from Parameterized Complexity Theory captures **the effectiveness of polynomial-time preprocessing** in a mathematically quantifiable manner. A **kernelization algorithm** for a parameterized problem with parameter k is a **polynomial-time** algorithm that converts any instance of the problem to an **equivalent** instance whose size is upper-bounded by a function $f(k)$ of the parameter k alone. The latter instance is called a **kernel** of size at most $f(k)$.

We show that the diverse variants of several basic problems, when parameterized by the solution size k and the number r of diverse solutions, admit kernels of **polynomial** size. In particular we show

Theorem

The following diverse subset minimization problems parameterized by $k + r$ admit polynomial kernels:

- **Diverse Vertex Cover**, on $\mathcal{O}(k(k + r))$ vertices;
- **Diverse d-Hitting Set** for fixed d, on $\mathcal{O}(k^d + kr)$ vertices;
- **Diverse Point Line Cover**, on $\mathcal{O}(k(k + r))$ points;
- **Diverse Feedback Arc Set in Tournaments**, on $\mathcal{O}(k(k + r))$ vertices.