

Push-pop walk orderings of graphs

Master 2 internship

1 Context

The internship belongs to the field of **enumeration** and **combinatorial Gray codes** [2]. An enumeration algorithm lists all solutions of a problem without repetition. Its efficiency is measured by its *delay*, the maximal time between two consecutive outputs. When solutions are large, or infinitely many, writing each of them in full requires an unbounded delay. The Gray code approach avoids this: each object is obtained from the previous one by a small change, and only the time to write the change is counted.

We apply this approach to the **enumeration of all walks of a directed graph**. The motivation comes from graph databases: modern query languages (Cypher, GQL, SQL/PGQ) extract walks satisfying navigational constraints. These walks are currently output in full, and output handling is sometimes the bottleneck. A possible solution is to output only the difference between consecutive walks.

A walk in a graph G is a sequence of consistent edges from G . A *push operation* adds an edge at either end of the walk; a *pop operation* removes the first or last edge. The *push-pop* distance between two walks is the minimal number of push and/or pop operations required to transform one walk into the other. A *walk ordering* of a graph G is a (generally infinite) sequence $w_1, \dots, w_n, \dots$ of walks in G such that all walks in G appears exactly once in the sequence. It is *k-bounded* if consecutive walks are at push-pop distance at most k , and *increasing* if w_{i+1} is never strictly shorter than w_i .

In a recent paper [1], we study the **pp-rank** of graph which is the least k such that the graph admits a k -bounded walk ordering. Similarly the **pp-increasing-rank** is the least k such that the graph admits a k -bounded increasing walk ordering. To this end, we introduce an easily computable graph parameter, the **peripherality** $\text{Peri}(G)$, which provides lower and upper bounds on the pp-increasing-rank of a graph. Our results are described in more details in Section 2, together with possible research directions.

2 Goals of the internship

The intern will first learn the basics of combinatorial Gray codes and the techniques of [1]. The paper leaves many open questions, some of which are listed below. **The intern will focus on only one or two of them**, according to their taste.

2.1 Tighter bounds on pp-rank

We know from [1] that the pp-rank is between 1 and 3. In particular, simple cycles have pp-rank 1 while the pp-rank of the directed path with five edges is already 2, the largest value known.



Figure 1: The directed path with five edges has a pp-rank 2

Question 1: We know that every inner-connected graph has pp-rank at most 3, but is there a graph of pp-rank 3?

Question 2: In the other direction, are all trees of pp-rank 2 or less? If yes, can we extend the result to all acyclic graphs?

Question 3: Is there an infinite family of acyclic graphs with a pp-rank of 1? We know only a few of such graphs, all with at most twelve edges, and one of them is given below

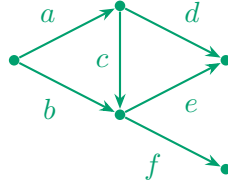


Figure 2: An acyclic graph with a pp-rank of 1

2.2 Tighter bound on pp-increasing-rank

We have shown in [1] that the pp-increasing-rank of a cyclic graph lies in $[\text{Peri}(G), 6\text{Peri}(G)]$, and we want to reduce the gap between lower and upper bounds. In the family below, the walk w_1aw_2 has peripherality $2k + 1$, and the pp-increasing-rank is exactly $2k + 2 = \text{Peri}(G) + 1$.

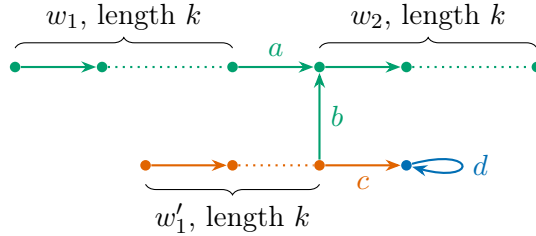


Figure 3: Peripherality $2k + 1$ and pp-increasing-rank $2k + 2$

Question 1: Is $\text{Peri}(G) + 1$, attained above, always a lower bound for pp-increasing rank?

Question 2: Find families with pp-increasing-rank of $6\text{Peri}(G)$, or improve the upper bound.

Since a push or pop always changes the length, it is tempting to think that the pp-increasing-rank is always even. However, it is not always the case: the graph below has pp-increasing-rank 3.

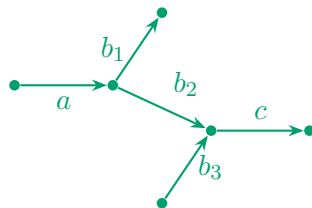


Figure 4: An acyclic graph with pp-increasing-rank 3

Question: Can we build graphs with an arbitrary odd pp-increasing-rank?

2.3 Computing the ranks

No algorithm is known to compute the pp-rank or the pp-increasing-rank (we do not even know whether they are computable). Of course, the difficulty comes from the fact that walk graphs are infinite and the naive approach would be to show that only walks of bounded length need to be considered. Conversely, one may look for hardness results for the pp-rank or the eventual pp-increasing-rank, by adapting the reduction of [1].

2.4 Levenshtein distance

We considered as elementary operations push and pops at either end of the walk. A natural extension would be to allow insertion and deletion at arbitrary position, that is consider the Levenshtein distance instead of the push-pop distance. Inner-connectedness remains necessary in that case, but peripherality no longer gives a lower bound on the increasing rank, thus new graph parameters are needed.

3 Continuation in PhD

The intern is welcome to continue in PhD with us, on the follow-up of this subject or a related topic. We don't have a specific funding for this PhD, but a good candidate is likely to obtain a funding from the graduate school.

4 Required skills

The topic requires few prerequisites, but a solid background in basic mathematics is essential: discrete mathematics, combinatorics, graph theory and writing rigorous proofs.

5 Supervision

- Supervisors: [Victor Marsault](#) (LIGM) and [Yann Strozecki](#) (DAVID).
The internship takes place at [LIGM](#), Université Gustave Eiffel, with possibility to work at the [DAVID](#) laboratory, Université de Versailles Saint-Quentin-en-Yvelines.
- Probable interactions with the other authors of [1], especially LIGM members Claire David and Nadime Francis.
- The supervisors will meet the intern several times a week.

References

- [1] Antoine Amarilli, Claire David, Nadime Francis, Victor Marsault, Mikaël Monet, and Yann Strozecki. Graphs with bounded Push-Pop walk orderings. ArXiv Preprint. <https://arxiv.org/abs/2610.19724>, 2026.
- [2] Torsten Mütze. Combinatorial Gray codes – an updated survey. *The Electronic Journal of Combinatorics*, 2023. <https://arxiv.org/abs/2202.01280>.