



Distance Labeling for Permutation Graphs

Fabrice Bazzaro¹ and Cyril Gavoille¹

*LaBRI UMR CNRS 5800, Université Bordeaux 1,
33405 Talence Cedex FRANCE.*

Abstract

We show that every permutation graph with n elements can be preprocessed in $O(n)$ time, if two linear extensions of the corresponding poset are given, to produce an $O(n)$ space data-structure supporting distance queries in constant time. The data-structure is localized and given as a distance labeling, that is each vertex receives a label of $O(\log n)$ bits so that distance queries between any two vertices are answered by inspecting on their labels only. This result improves the previous scheme due to Katz, Katz and Peleg [20] in the STACS '00 by a $\log n$ factor.

Keywords: algorithms, data-structure, distributed algorithms, distance queries, distance labeling scheme, permutation graphs

1 Introduction

The comparability graphs of two-dimensional posets are exactly the permutation graphs, namely the intersection graphs of straight segments between two parallel lines [4]. Intersection graphs are graphs in which vertices are mapped to objects, with the vertices defined to be adjacent if and only if the corre-

¹ {bazzaro,gavoille}@labri.fr

The two authors are supported by the project “PairAPair” of the ACI Masses de Données.

sponding objects have nonempty intersection. See [23] for a comprehensive introduction to the intersection graphs.

This paper deals with the problem of the distance computation and distributed abilities of permutation graphs. Commonly, when we make a query concerning a set of nodes in a graph (adjacency, distance, connectivity, etc.), we need to make a global access to the structure. In our approach, the compromise is to store the maximum of information in a label associated with a vertex to have directly what we need with a local access. Motivation of localized data-structures in distributed computing is survey and widely discussed in [17].

We are especially interested in the distance labeling problem, introduced in [24]. The problem consists in labeling the vertices of a graph to compute the distance between any two of its vertices x and y using only the information stored in the labels of x and y , without any other source of information. The main parameters taken into account when designing a solution are: (1) length (in bits) of the labels; (2) time complexity to decode the distance from the labels; and (3) time complexity to preprocess the graph and to compute all the labels.

Related works

Distance computation in graphs is one of the most fundamental graph algorithmic problem. Computing the distance matrix of a general graph is strongly related to Boolean matrix multiplication [12], and achieving this task as quickly as possible is a widely open problem.

However, the time complexity of this problem is known, and can be reduced significantly from the naive $O(n^3)$ upper bound, for many families of graphs: planar graphs [11,21,29], bounded tree-width graphs [7], interval graphs [3,8,16], etc.

Beyond the classical all-pair distance problem, whose goal is to preprocess (possibly linearly) a graph and to produce a data-structure supporting distance or shortest-path queries in the minimum time complexity, the distance labeling problem is a variant in which the queries must be answered *locally*, by looking at the information related to the concerned vertices only. Introduced in [24], it generalizes adjacency labeling [1,19] whose goal is only to decide whether the distance is 1 or not between any two vertices. At this point, it is worth to mention that any distance labeling scheme on a family \mathcal{F} of graphs with $\ell(n)$ bit labels converts trivially into a non-distributed data-structure for \mathcal{F} of $O(\ell(n) \cdot n / \log n)$ space supporting distance queries within the same time

complexity, being assumed that a cell of space can store $\Omega(\log n)$ bits of data.

The main results on the field are that general graphs support a distance labeling scheme with labels of $O(n)$ bits [18], and that trees [1,24], bounded tree-width graphs [18], distance-hereditary graphs [15], bounded clique-width graphs [10], some non-positively curved plane graphs [9], interval and permutation graphs, all support distance labeling schemes with $O(\log^2 n)$ bit labels. There are also deep results concerning approximated distance labeling schemes we will not discuss here, e.g., see [14,27,29,30].

The $O(n)$ bit upper bound is tight for general graphs, and a lower bound of $\Omega(\log^2 n)$ bit on the label length is known for trees [18], implying that all the results mentioned above are tight as well, excepted for interval and permutation graphs that does not contain trees. Recently, [16] showed an optimal bound of $O(\log n)$ bits for interval graphs and circular-arc graphs.

Concerning permutation graphs, we should mention the related work of [25]. It is shown how to compute one fixed BFS tree and DFS tree in $O(n)$ and $O(n \log \log n)$ time respectively when the permutation of the graph is given. Although original, the technique is limited; it does not allow us to choose arbitrarily the root of the trees.

2 Our results

We show that permutation graphs with n vertices enjoy a distance labeling with labels of length $O(\log n)$ bits. The distance can be computed in constant time, and all the labels are computed in $O(n)$ time if a realizer (the permutation associated to the representation of the graph as an intersection model of straight segments between two parallel lines) is given. (Such realizer can be obtained in $O(n+m)$ time [22] otherwise, m the number of edges). This result has optimal complexities (label length, distance decoder time complexity, and preprocessing time complexity). It improves within a $\log n$ factor on the label length the previous result of Katz, Katz and Peleg [20] in the STACS '00.

As remarked previously, this leads to an optimal $O(n)$ space data-structure supporting distance queries in constant time, and computable in $O(n)$ time.

Outline of our distance labeling scheme

We use a 2D geometric representation of the permutation graph, each vertex being associated with a point of \mathbb{N}^2 , and u is adjacent to v if and only if v is in the South-East or North-West quadrant around u . For our purpose, two sets of points (or vertices) are of special interests: those with no neighbors

in their North-West quadrant (call A), and those with no neighbors in their South-East quadrant (call B). Intuitively, one can always construct a shortest path between non-adjacent vertices by using only edges alternating between A and B . The main difficulty resides in deciding whether the first edge must be incident to a vertex of A or of B .

The first trick is to treat differently short and long distances. We entirely characterize distances ≤ 3 by comparing the coordinates of six specific vertices spread around each vertex (three belongs to A and three to B). Then, for long distances, i.e., distances ≥ 4 , we show that they reduce to the distance computation between vertices of two intermediate graphs, G_A and G_B , defined on respectively the vertices of A and of B . The edges of G_A and G_B are entirely determined by an asymmetric relation between the points of A and of B , and we show that a distance labeling for these graphs with $O(\log n)$ bit labels is possible. The distance is then computed by evaluating the distance in the graphs G_A and G_B between the six points associated with the source and the six points associated with the destination.

Finally, after several optimizations, the resulting data-structure is extremely simple. It is composed of 9 integers of $\{0, \dots, n-1\}$ plus 6 bits. The distance decoder simply consists of a constant number of additions and comparisons on these integers.

Theorem 2.1 *Permutation graphs with n vertices enjoy a distance labeling scheme using labels of $9 \lceil \log n \rceil + 6$ bits. The distance decoder has constant time complexity, and given a realizer of the graph, i.e., a permutation of $\{1, \dots, n\}$, all the labels can be computed in $O(n)$ time.*

3 Conclusion and Discussion

An interesting direction for further research would be to discover other large families of graphs supporting distance labeling schemes with logarithmic or poly-logarithmic labels (in n , the number of vertices). To our best knowledge, only interval and permutation graphs (and their immediate generalizations into circular-arc and circular permutation graphs) have $O(\log n)$ bit label schemes. Trees, bounded tree-width and bounded clique-width graphs have $O(\log^2 n)$ bit distance labeling schemes.

As a byproduct, we also show how to adapt our data-structure such that a length- d shortest path can be extracted in $O(d)$ time for any distance- d pair of vertices. One can also use our localized data-structure to identify the first shortest-path edge, and we therefore present (cf. [5,6]) a new shortest-path

compact routing scheme for permutation graphs, improving the result of [13]. All these results can be extended to circular permutation graphs, a natural generalization of permutation graphs.

Intersection graphs, like interval or permutation graphs, seem a priori good candidates for this search, essential because such graphs are “simple” in the Kolmogorov Complexity sense: they support short programs to decide adjacency. However we moderate this feeling because many intersection graph families contain planar graphs. Although planar graphs have also short adjacency labels, $3 \log n + O(\log^* n)$ bits per vertex from [2], it is known that every distance labeling scheme on this family requires some labels of $\Omega(n^{1/3})$ bits [18]. Some generalizations of interval graphs, namely interval number-3 graphs (the intersection graphs of the union of 3 intervals) and boxicity-3 graphs (the intersection graphs of the Cartesian product of 3 intervals), both contain planar graphs [26,28]. So, the design of distance labeling with $O(\log n)$ bit labels (and related problems like all-pair shortest path and compact routing) remains open for these generalizations only for boxicity-2, and interval number-2 graphs.

Another direction is to extend our data-structure in a dynamic setting.

References

- [1] S. ALSTRUP, P. BILLE, AND T. RAUHE, *Labeling schemes for small distances in trees*, in 14th Symposium on Discrete Algorithms (SODA), ACM-SIAM, Jan. 2003, pp. 689–698.
- [2] S. ALSTRUP AND T. RAUHE, *Small induced-universal graphs and compact implicit graph representations*, in 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society Press, Nov. 2002, pp. 53–62.
- [3] M. J. ATALLAH, D. Z. CHEN, AND D. T. LEE, *An optimal algorithm for shortest paths on weighted interval and circular-arc graphs, with applications*, *Algorithmica*, 14 (1995), pp. 429–441.
- [4] K. A. BAKER, P. C. FISHBURN, AND F. S. ROBERTS, *Partial orders of dimension 2*, *Networks*, 2 (1971), pp. 11–28.
- [5] F. BAZZARO AND C. GAVOILLE, *Localized and compact data-structure for comparability graphs*, Research Report RR-1343-05, LaBRI, University of Bordeaux 1, 351, cours de la Libération, 33405 Talence Cedex, France, Feb. 2005.

- [6] ———, *Localized and compact data-structure for comparability graphs*, in 16th Annual International Symposium on Algorithms and Computation (ISAAC), vol. Lecture Notes in Computer Science, Springer, Dec. 2005. To appear.
- [7] S. CHAUDHURI AND C. D. ZAROLIAGIS, *Shortest paths in digraphs of small treewidth. Part I: Sequential algorithms*, *Algorithmica*, 27 (2000), pp. 212–226.
- [8] D. Z. CHEN, D. T. LEE, R. SRIDHAR, AND C. N. SEKCHARAN, *Solving the all-pair shortest path query problem on interval and circular-arc graphs*, *Networks*, 31 (1998), pp. 249–257.
- [9] V. D. CHEPOI, F. F. DRAGAN, AND Y. VAXES, *Distance and routing labeling schemes for non-positively curved plane graphs*, *Journal of Algorithms*, (2005). To appear.
- [10] B. COURCELLE AND R. VANICAT, *Query efficient implementation of graphs of bounded clique-width*, *Discrete Applied Mathematics*, 131 (2003), pp. 129–150.
- [11] H. N. DJIDJEV, *Efficient algorithms for shortest path queries in planar digraphs*, in 22nd International Workshop on Graph - Theoretic Concepts in Computer Science (WG), vol. 1197 of Lecture Notes in Computer Science, Springer, June 1996, pp. 151–165.
- [12] D. DOR, S. HALPERIN, AND U. ZWICK, *All-pairs almost shortest paths*, *SIAM Journal on Computing*, 29 (2000), pp. 1740–1759.
- [13] F. F. DRAGAN AND I. LOMONOSOV, *New routing schemes for interval graphs, circular-arc graphs, and permutation graphs*, in 14th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS), Nov. 2002, pp. 78–83.
- [14] C. GAVOILLE, M. KATZ, N. A. KATZ, C. PAUL, AND D. PELEG, *Approximate distance labeling schemes*, in 9th Annual European Symposium on Algorithms (ESA), F. M. auf der Heide, ed., vol. 2161 of Lecture Notes in Computer Science, Springer, Aug. 2001, pp. 476–488.
- [15] C. GAVOILLE AND C. PAUL, *Distance labeling scheme and split decomposition*, *Discrete Mathematics*, 273 (2003), pp. 115–130.
- [16] ———, *Optimal distance labeling schemes for interval and circular-arc graphs*, in 11th Annual European Symposium on Algorithms (ESA), G. D. Battista and U. Zwick, eds., vol. 2832 of Lecture Notes in Computer Science, Springer, Sept. 2003, pp. 254–265.
- [17] C. GAVOILLE AND D. PELEG, *Compact and localized distributed data structures*, *Journal of Distributed Computing*, 16 (2003), pp. 111–120. PODC 20-Year Special Issue.

- [18] C. GAVOILLE, D. PELEG, S. PÉRENNÈS, AND R. RAZ, *Distance labeling in graphs*, *Journal of Algorithms*, 53 (2004), pp. 85–112.
- [19] S. KANNAN, M. NAOR, AND S. RUDICH, *Implicit representation of graphs*, *SIAM Journal on Discrete Mathematics*, 5 (1992), pp. 596–603.
- [20] M. KATZ, N. A. KATZ, AND D. PELEG, *Distance labeling schemes for well-separated graph classes*, in 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS), H. Reichel and S. Tison, eds., vol. 1770 of *Lecture Notes in Computer Science*, Springer Verlag, Feb. 2000, pp. 516–528.
- [21] P. N. KLEIN, S. RAO, M. R. HENZINGER, AND S. SUBRAMANIAN, *Faster shortest-path algorithms for planar graphs*, *Journal of Computer and System Sciences*, 55 (1997), pp. 3–23.
- [22] R. M. MCCONNELL AND J. P. SPINRAD, *Linear-time transitive orientation*, in 8th Symposium on Discrete Algorithms (SODA), ACM-SIAM, Jan. 1997, pp. 19–25.
- [23] T. A. MCKEE AND F. R. MCMORRIS, *Topics in Intersection Graph Theory*, SIAM Monographs on Discrete Mathematics and Applications, 1999.
- [24] D. PELEG, *Proximity-preserving labeling schemes*, *Journal of Graph Theory*, 33 (2000), pp. 167–176.
- [25] C. J. RHEE, Y. D. LIANG, S. K. DHALL, AND S. LAKSHMIVARAHAN, *Efficient algorithms for finding depth-first and breadth-first search trees in permutation graphs*, *Information Processing Letters*, 49 (1994), pp. 45–50.
- [26] E. R. SCHEINERMAN AND D. B. WEST, *Interval number of a planar graph: Three intervals suffice*, *Journal of Combinatorial Theory, Series B*, 35 (1983), pp. 224–239.
- [27] K. TALWAR, *Bypassing the embedding: Algorithms for low dimensional metrics*, in 36th Annual ACM Symposium on Theory of Computing (STOC), ACM Press, June 2004, pp. 281–290.
- [28] C. THOMASSEN, *Interval representations of planar graphs*, *Journal of Combinatorial Theory, Series B*, 40 (1986), pp. 9–20.
- [29] M. THORUP, *Compact oracles for reachability and approximate distances in planar digraphs*, in 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society Press, Oct. 2001, pp. 242–251.
- [30] M. THORUP AND U. ZWICK, *Approximate distance oracles*, in 33rd Annual ACM Symposium on Theory of Computing (STOC), ACM Press, July 2001, pp. 183–192.