Local 3-approximation algorithms for weighted dominating set and vertex cover in quasi unit-disk graphs

Marja Hassinen, Valentin Polishchuk, and Jukka Suomela
Helsinki Institute for Information Technology HIIT,
Department of Computer Science, University of Helsinki,
P.O. Box 68, FI-00014 University of Helsinki, Finland
{firstname.lastname}@cs.helsinki.fi

***Abstract*—We present a simple 3-approximation algorithm for minimum-weight dominating set and minimum-weight vertex cover in unit-disk graphs and quasi unit-disk graphs in which each node knows its coordinates. The algorithm is local: the output of a node depends solely on the input within its constant-radius neighbourhood. The local horizon of the algorithm is small, both in the worst case and on average.**

I. Introduction

Data propagation in wireless and ad-hoc sensor networks is subject to link failures, interference, and delays. It makes gathering full information about a large-scale network in one place an unrealistic task, especially in a dynamic setting when the nodes may sporadically join and leave the network. This calls for designing algorithms whose output at each node of the network depends only on the local neighbourhood of the node — the so called "local" algorithms.

Formally, an algorithm is *local* if there is a constant r such that the decision of any node is a function the inputs at nodes within r or fewer edges (hops) from the node; the constant r is called the *local horizon* (or locality distance) of the algorithm. In other words, a local algorithm is equal to a constant-time algorithm in Linial's [1] and Naor and Stockmeyer's [2] model: we allow only r communication steps, but message size is unbounded and local computation is free. Unlike Linial and Naor and Stockmeyer, we assume that each node knows its coordinates.

We work in the *unit-disk graph* model where there is an edge (communication link) between two nodes whenever the nodes are within distance 1 from each other. We consider finding a minimum-weight dominating set and a minimum-weight vertex cover in unit-disk graphs. A *dominating set* is a subset of nodes such that every node is either in the set or is connected to one of the nodes in the set. A *vertex cover* is a subset of nodes such that every edge is incident to at least one node in the cover. Finding dominating sets or vertex covers of small size allows one to group nodes or links of the network into "clusters", with only one node in a cluster responsible for communication with the nodes in the cluster. This leads to energy conservation and interference reduction.

The performance of a local algorithm is measured by comparing the size (or the weight, in the weighted version of the problem) of its output to that of the optimal solution. Specifically, the (worst-case) *approximation ratio* of the algorithm is the supremum, over all problem instances, of the ratio of the size of the solution produced by the algorithm to the size of the optimal solution. Note that since the problems that we consider are NP-hard, even in a centralised setting only approximate solutions can be found in polynomial time (unless P $=$ NP).

A. Prior Work

Urrutia [3] presents a local, factor 15 approximation algorithm for dominating set in unit-disk graphs. Czyzowicz et al. [4] present a local factor 5 approximation. Wiese and Kranakis [5] present a local approximation scheme, that is, a local $(1+\epsilon)$-approximation algorithm for any $\epsilon > 0$.

There is a trivial, local, factor 12 approximation algorithm for vertex cover in unit-disk graphs: pick all vertices [6], [7]. Wiese and Kranakis [7] present a local approximation scheme.

In the context of the dominating set problem, Wiese and Kranakis [5] raise the question of what is the smallest local horizon r with which we are able to achieve a given approximation ratio. The 5-approximation algorithm [4] has local horizon $r = 11$. The local horizon of the approximation scheme [5] depends on the desired approximation ratio; for example, to obtain a 3-approximation, $r = 46814$ is sufficient.

There is related work on local algorithms beyond unit-disk graphs as well. Kuhn and Wattenhofer [8] present a randomised local algorithm for bounded-degree graphs; the algorithm finds a dominating set with expected size at most a nontrivial constant factor times the optimum. Wiese and Kranakis [7] suggest to study local algorithms for vertex cover and related problems in quasi unit-disk graphs [9].

We emphasise that our definition of local algorithms requires a strictly constant local horizon r. For example, the approximation scheme by Kuhn et al. [10] is not local in this strict sense.

B. Contributions

We present a simple, local, factor 3 approximation algorithm for minimum-weight dominating set and minimum-weight vertex cover in unit-disk graphs. The local horizon is $r = 83$. The algorithm also works in quasi unit-disk graphs for which we obtain the same approximation factor 3, at the cost of having a larger local horizon.

In addition to minimum-weight dominating set and vertex cover, the same idea can be applied to various other covering problems.

As with other local algorithms, a local change in network topology affects our solution only locally — within at most 83 hops. In particular, if a dominator node leaves the network, the nodes that were uniquely dominated by the dominator can resolve the problem in their (common) local neighbourhood in constant time (under the assumption that local computations are free).

The approximation factor of 3 and the local horizon $r = 83$ of our algorithm are due to the partitioning of the plane into 2×4 rectangles of three colours, with same colour rectangles sufficiently separated. Making the rectangles smaller would require more colours to ensure separation, leading to a higher approximation factor. On the other hand, making the rectangles larger would result in a larger local horizon (while we would still need 3 colours, i.e., the approximation factor would not improve).

II. Minimum-Weight Dominating Set

We present an algorithm for approximating the minimum-weight dominating set. The input of a node consists of its identifier, its coordinates, and the list of its neighbours. Each node must decide, based only on input available within its local horizons, whether it participates in the solution. We prove that the constructed solution is a feasible dominating set and its weight is at most 3 times the optimum.

The algorithm divides the problem into subproblems based on the locations of the nodes and solves the subproblems optimally. The algorithm's solution is the union of the subproblem solutions.

Step 1, construct a plane subdivision: Divide the plane into rectangles as shown in Figure 1a. The rectangles are 2 units wide and 4 units high. We can choose that the upper and left borders (including the upper right corner) are part of a rectangle but the lower and right borders (including the lower left corner) are not. Each node knows its coordinates, so it can calculate in which rectangle it is.

Each rectangle is assigned a colour (1, 2, or 3) as shown. The colours are chosen so that rectangles having the same colour are not next to each other.

Then we construct "extended rectangles", obtained from the 2×4 rectangles by adding points whose distance from the rectangle is at most 1 (see Figure 1b). Extended rectangles are round-corner rectangles of size 4×6. The extensions of two rectangles with the same colour do not overlap. Each point in the plane belongs to at most 3 extended rectangles, one for each colour.

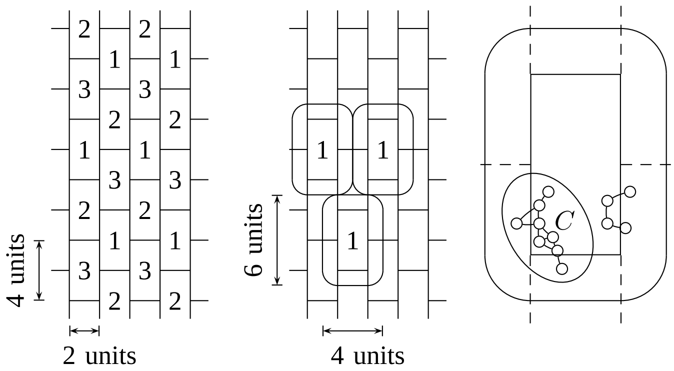

Fig. 1. (a) Plane subdivision. (b) Extended rectangles. (c) Graph induced by nodes within an extended rectangle, and its connected component C.

Consider the graph induced by the nodes within an extended rectangle, as shown in Figure 1c. The obtained subgraph may consist of several connected components. The diameter of each connected component is limited by a constant because the shortest path between two nodes cannot be arbitrarily long if it lies inside an extended rectangle (see Section V). Therefore the connected component fits inside the radius r neighbourhood of each of its nodes, if r is chosen to be large enough.

Step 2, formulate the subproblems: Each connected component constitutes a subproblem which is solved locally by its nodes. The nodes find a minimum-weight subset which dominates all nodes inside the original 2×4 rectangle. Nodes outside the 2×4 rectangle but inside the extended rectangle can participate in dominating other nodes but do not need to be dominated themselves.

We can assume that each node gathers full information about its connected component and then applies a deterministic algorithm to compute the solution. Thus, all nodes of a connected component agree on which nodes constitute the local solution.

Step 3, combine the local solutions: The algorithm's output is the set of nodes which have, at least once, participated in a local solution. This is a feasible dominating set: each node belongs to exactly one 2×4 rectangle and is thus dominated by a subproblem solution.

III. Minimum-Weight Vertex Cover

We apply the same idea for approximating the minimum-weight vertex cover. Again, each node must decide, based on local information, whether it participates in the solution.

Step 1, construct a plane subdivision: The plane subdivision, extended rectangles, and connected components are constructed the same way as in Section II.

Step 2, formulate the subproblems: Each connected component constitutes a subproblem which is solved locally by its nodes. The nodes find a minimum-weight subset which covers all edges that have at least one endpoint inside the original 2×4 rectangle. Edges whose both endpoints are inside the extended rectangle but not inside the original rectangle are not considered.

Step 3, combine the local solutions: The algorithm's solution is the set of nodes which have, at least once, participated in a local solution. The produced set is a feasible vertex cover: the endpoints of an edge lie inside one or two 2×4 rectangles and the edge is covered in the subproblem solutions related to those rectangles.

IV. Approximation Guarantee

The approximation ratios for minimum-weight dominating set and minimum-weight vertex cover are analysed the same way. Consider a minimum-weight solution D and let $w(D)$ be its weight. For each subproblem, intersection of the set D and the extended rectangle provides a feasible solution. Extended rectangles of colour 1 do not overlap; therefore the total weight of the optimal solutions of subproblems of colour 1 is at most $w(D)$. The same applies to subproblems of colour 2 and colour 3. The algorithm's solution is the union of the subproblem solutions and has weight at most $3w(D)$.

V. Local Horizon

In unit-disk graphs, the diameter of a connected component inside an extended rectangle is limited by a constant. To see this, consider two arbitrary nodes, u and v, and the shortest path between them. Take every second node on the path. Draw disks of radius $1/2$ centred on these nodes. The disks are pairwise-disjoint, since otherwise the path would not be the shortest one.

The centres of the disks lie inside a 4×6 round-corner rectangle, and thus the disks themselves lie inside a 5×7 round-corner rectangle. The total area of the disks may not exceed the area of the rectangle. This means that no more than 42 disks can be packed inside the rectangle, which implies that the shortest path between u and v can consist of at most 84 nodes (including u and v), and hence the diameter of a connected component is at most 83.

We conducted simulations to estimate the diameter in the average case as well. For each value of $n = 1, 2, \dots, 200$, we ran 50000 experiments. In each experiment, we created a unit disk graph by placing n nodes uniformly at random within a 4×6 rectangle. Then we computed the diameter of the graph, taking the maximum over all connected components. The results are reported in Figure 2. The thick solid line is the average value; the thin solid lines indicate the region with 99% of the values; and grey vertical bars show the full range of the values that we observed in the experiments. We never observed the diameter to be larger than 25. As the number of points grows, the diameter approaches 8, the diameter of the rectangle. As a further confirmation of this, we ran 100 experiments for each value of $n = 1100, 1200, \dots, 10000$; in each of these cases, the diameter was 8.

The dashed line in Figure 2 shows the fraction of graphs that consisted of one connected component. Naturally all graphs are connected in the case $n = 1$, and almost all graphs are connected for large values of n, but for intermediate values, most networks consist of several connected components.

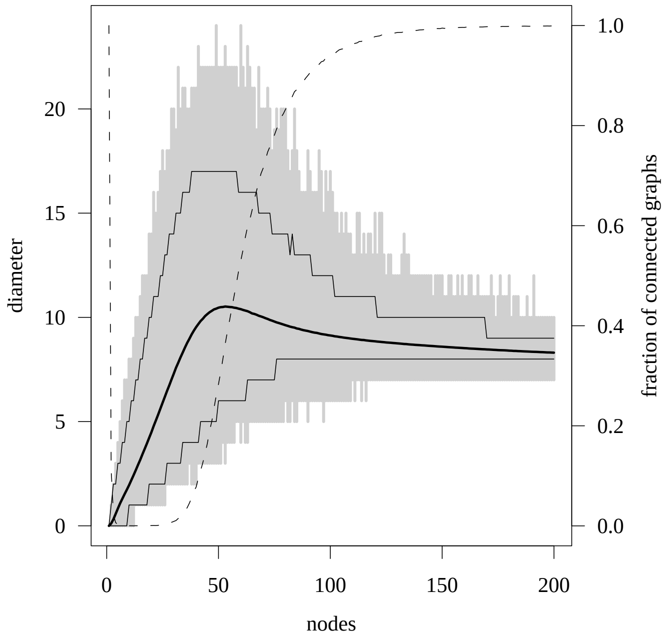

Fig. 2. Diameter of a connected component (solid lines, left axis; the thick line is the average and the thin lines indicate the 99% region), and the fraction of networks that consist of one connected component (dashed line, right axis).

VI. Running Time

Our algorithm is local, in the strict sense that the output of a node depends solely on the input within its constant-radius neighbourhood. In bounded-degree graphs, a local algorithm is a linear-time centralised algorithm as well: there is a constant upper bound on the number of nodes in the local neighbourhood of each node, and therefore each local computation can be performed in constant time. This is not the case if the degree of the graph is unbounded; a local algorithm is not necessarily a polynomial-time centralised algorithm. Our simple local algorithm does not make obsolete, for example, Ambühl et al.'s [11] polynomial-time algorithm for approximating minimum-weight dominating set in unit-disk graphs.

Nevertheless, if we focus on *unweighted* dominating set and vertex cover, each subproblem in our algorithm can be solved in polynomial time [12]. The nodes in the subproblem all lie within a constant-area region in the plane. For example, by a packing argument, the cardinality of an independent set has a constant upper bound. A maximal independent set is a feasible dominating set, and thus a minimum-size (but not minimum-weight) dominating set can be found by a polynomial-time brute-force algorithm.

In Sections II and III, we assumed that each node solves the subproblem for its connected component. We can avoid redundant work at the expense of more communication as follows. Nodes in the connected component choose a leader. The leader computes the optimal solution, and informs all nodes of whether they participate in the local solution. Gathering information and sending the results requires transferring data over the distance of at most $2r$ hops.

VII. Quasi Unit-Disk Graphs

Our approach can be applied to d-quasi unit-disk graphs [9] for any $0 < d \leq 1$. The two properties of unit-disk graphs utilised by the algorithm are that (i) no edges longer than 1 exist and (ii) the diameter of a connected component inside an extended rectangle is limited. For a small d, we need a larger local horizon, but we can still achieve the approximation ratio 3.

Acknowledgements

We thank Evangelos Kranakis, Andreas Wiese, Roger Wattenhofer, and Petteri Kaski for comments and discussions, and anonymous reviewers for their helpful feedback. This research was supported in part by the Academy of Finland, Grant 116547, and by Helsinki Graduate School in Computer Science and Engineering (Hecse).

References

[1] N. Linial, "Locality in distributed graph algorithms," *SIAM Journal on Computing*, vol. 21, no. 1, pp. 193–201, 1992.

[2] M. Naor and L. Stockmeyer, "What can be computed locally?" *SIAM Journal on Computing*, vol. 24, no. 6, pp. 1259–1277, 1995.

[3] J. Urrutia, "Local solutions for global problems in wireless networks," *Journal of Discrete Algorithms*, vol. 5, no. 3, pp. 395–407, 2007.

[4] J. Czyzowicz, S. Dobrev, T. Fevens, H. González-Aguilar, E. Kranakis, J. Opatrny, and J. Urrutia, "Local algorithms for dominating and connected dominating sets of unit disk graphs with location aware nodes," in *Proc. 8th Latin American Theoretical Informatics Symposium (LATIN, Búzios, Brazil, April 2008)*, ser. Lecture Notes in Computer Science, vol. 4957. Berlin, Germany: Springer-Verlag, 2008, pp. 158–169.

[5] A. Wiese and E. Kranakis, "Local PTAS for dominating and connected dominating set in location aware unit disk graph," Carleton University, School of Computer Science, Ottawa, Canada, Tech. Rep. TR-07-17, Oct. 2007.

[6] F. Kuhn, "The price of locality: Exploring the complexity of distributed coordination primitives," Ph.D. dissertation, ETH Zürich, Dec. 2005.

[7] A. Wiese and E. Kranakis, "Local PTAS for independent set and vertex cover in location aware unit disk graphs," in *Proc. 4th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS, Santorini Island, Greece, June 2008)*. Berlin, Germany: Springer-Verlag, 2008, to appear.

[8] F. Kuhn and R. Wattenhofer, "Constant-time distributed dominating set approximation," *Distributed Computing*, vol. 17, no. 4, pp. 303–310, 2005.

[9] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad hoc networks beyond unit disk graphs," *Wireless Networks*, 2007, to appear.

[10] F. Kuhn, T. Nieberg, T. Moscibroda, and R. Wattenhofer, "Local approximation schemes for ad hoc and sensor networks," in *Proc. Joint Workshop on Foundations of Mobile Computing (DIALM-POMC, Cologne, Germany, September 2005)*. New York, NY, USA: ACM Press, 2005, pp. 97–103.

[11] C. Ambühl, T. Erlebach, M. Mihal'ák, and M. Nunkesser, "Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs," in *Proc. 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 10th International Workshop on Randomization and Computation (APPROX and RANDOM, Barcelona, Spain, August 2006)*, ser. Lecture Notes in Computer Science, vol. 4110. Berlin, Germany: Springer-Verlag, 2006, pp. 3–14.

[12] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns, "NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs," *Journal of Algorithms*, vol. 26, no. 2, pp. 238–274, 1998.