NOTICE: This is the author's version of a work that was accepted for publication in Information Processing Letters. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Information Processing Letters, Volume 103, Issue 1, 30 June 2007, pages 28–33.
doi:10.1016/j.ipl.2007.02.001

Approximability of Identifying Codes and Locating-Dominating Codes

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

Abstract

We study the approximability and inapproximability of finding identifying codes and locating-dominating codes of the minimum size. In general graphs, we show that it is possible to approximate both problems within a logarithmic factor, but sublogarithmic approximation ratios are intractable. In bounded-degree graphs, there is a trivial constant-factor approximation algorithm, but arbitrarily low approximation ratios remain intractable. In so-called local graphs, there is a polynomial-time approximation scheme. We also consider fractional packing of codes and a related problem of finding minimum-weight codes.

Key words: approximation algorithms, combinatorial problems, graph algorithms, identifying codes, locating-dominating codes.

1 Introduction

Consider an undirected graph $G = (V, E)$ and a matrix $H = (h_{uv})$ of size $|V| \times |V|$ where h_{uv} is the *proximity* of vertex v as seen from vertex u. Each subset $C \subseteq V$ determines a matrix $H(C)$ of size $|V| \times |C|$ which is formed by restricting to the columns $v \in C$. We say that C is a *code* for this proximity matrix H if the rows of $H(C)$ are distinct and each row contains a nonzero

Email address: `jukka.suomela@cs.helsinki.fi` (Jukka Suomela).

element. The vertices of the code C are called *beacons*. We can determine our location in the graph by measuring our proximity to each beacon; as the rows of $H(C)$ are distinct, the proximity information uniquely identifies the vertex.

In this paper, we study the *optimisation problems of finding a code of the minimum size*. We focus on the following definitions of proximity. We write $d(u, v)$ for the shortest-path distance (the number of edges) from vertex u to vertex v. For t-IDENTIFYING CODE (t-IC), set $h_{uv} = 1$ if $d(v, u) \leq t$; otherwise $h_{uv} = 0$. For t-LOCATING-DOMINATING CODE (t-LDC), set $h_{uv} = 2$ if $u = v$; otherwise set $h_{uv} = 1$ if $d(v, u) \leq t$; otherwise set $h_{uv} = 0$.

Note that a t-LDC always exists, as we can choose $C = V$. This is not necessarily the case for t-IC. However, if a code exists, $C = V$ is a code. As it is easy to test whether there is a code for a given graph, we focus on graphs where a code exists.

Motivation. Consider the problem of installing devices such as motion detectors. The vertices V correspond to physical areas, e.g., rooms; the edges E describe the ability to detect events in neighbouring areas, e.g., a line of sight; and a beacon $c \in C$ corresponds to an area equipped with a detector. The goal is to determine in which room there is motion, assuming there is at most one such room. If each detector is a three-state device that is able to distinguish between no event, an event in a neighbouring vertex, and an event in its own vertex, we arrive at the 1-LDC formulation [20]. If each detector is a two-state device that cannot distinguish between events in its own vertex and in a neighbouring vertex, we arrive at the 1-IC formulation [10]. If we were only interested in determining whether there is motion somewhere in the building (instead of locating the room in which there is motion), it would suffice to consider sets C which are dominating sets of G.

Related work. The problems t-IC and t-LDC are known to be NP-complete for all $t \geq 1$, in both directed and undirected graphs [1,2,4,5]. Extensive research has been conducted on identifying codes and locating-dominating codes in specific graphs and restricted problem classes, such as strips, square lattices, hexagonal lattices, triangular lattices, king lattices, Hamming spaces, chains, cycles, trees, and series-parallel graphs; see Lobstein [16] for an online bibliography. However, little is known about the approximability of t-IC and t-LDC in more general problem classes [18, §4.1]. The related problems METRIC DIMENSION and ALARM PLACEMENT are known to be approximable within a logarithmic factor [12,15].

Contributions. In Section 2, we study the approximability of 1-IC and 1-LDC in *general graphs*. We prove that it is possible to approximate both problems within a logarithmic factor, but (under plausible complexity-theoretic assumptions) sublogarithmic factors are intractable. In Section 3, we consider 1-IC and 1-LDC in *graphs of bounded degree*. We show that there is a trivial constant-factor approximation algorithm, but approximating 1-IC or 1-LDC within an arbitrarily low constant factor is intractable. In Section 4, we focus on the class of so-called *local graphs* that are motivated by practical applications. We prove that in these cases, t-IC and t-LDC admit a polynomial-time approximation scheme (PTAS). In Section 5, we conclude the paper by having a look at two related problems: *fractional packing of codes* and *minimum-weight codes*.

2 Approximability in General Graphs

We first prove that t-IC and t-LDC can be approximated within a logarithmic factor. We use the same general approach as Khuller et al. [12] for METRIC DIMENSION and Lakshmanan et al. [15] for ALARM PLACEMENT: we construct an equivalent instance of SET COVERING.

We write $X \diamond Y$ for the set of all unordered pairs $\{x, y\}$ where $x \in X$, $y \in Y$, and $x \neq y$. We formulate t-IC and t-LDC equivalently in terms of finding beacons that *cover* all vertices and *distinguish* all vertex pairs. For each beacon c, the set of covered vertices, $S(c)$, consists of all vertices $u \in V$ such that $h_{uc} \neq 0$. The set of distinguished vertex pairs, $T(c)$, consists of all pairs $\{u, v\} \in V \diamond V$ such that $h_{uc} \neq h_{vc}$.

By definition, a subset of vertices is a code if and only if each vertex is covered by at least one beacon and each vertex pair is distinguished by at least one beacon. Finding a t-IC or t-LDC of size k is thus equivalent to finding k sets $S(c) \cup T(c)$ such that their union equals $V \cup (V \diamond V)$; this is an instance of SET COVERING, which can be approximated within a logarithmic ratio in polynomial time by a greedy algorithm [9].

We next prove that this ratio is asymptotically tight for $t = 1$; we use similar ideas as in the proof of the inapproximability of ALARM PLACEMENT [15]. Consider an instance of DOMINATING SET. Given a graph $G' = (V', E')$, the goal is to find a minimum subset of vertices $X \subseteq V'$ such that each $i \in V' \setminus X$ has a neighbour in X. We assume $|V'| \geq 2$, as small instances are trivial.

Let $2V' = \{1, 2\} \times V'$ and form any injection $f \colon V' \times V' \to 2V' \diamond 2V'$. Construct a graph $G = (V, E)$ as follows. The set of vertices V consists of p_{ki}, q_{ki}, r_{ki}, ϕ_a, and ψ_a for all $k \in V'$, $i \in V'$, and $a \in 2V'$. The set of edges E

consists of $\{p_{ki}, \phi_a\}$, $\{p_{ki}, \phi_b\}$, $\{q_{ki}, \phi_a\}$, $\{q_{ki}, \phi_b\}$, $\{r_{ki}, \psi_a\}$, and $\{r_{ki}, \psi_b\}$ for all $f(k,i) = \{a,b\}$; and $\{p_{ki}, r_{kj}\}$ for all i, j, k such that either $i = j$ or $\{i,j\} \in E'$. See Fig. 1 for an illustration.

Let X be a dominating set in G'. Construct $C = \{r_{ki} : k \in V', i \in X\} \cup \{\phi_a, \psi_a : a \in 2V'\}$. The size of C is $|X||V'| + 4|V'|$. This set is a 1-IC and a 1-LDC, as it covers V and distinguishes all $V \diamond V$ in both formulations.

Conversely, let C be a 1-IC or a 1-LDC for G. Construct $|V'|$ sets $X_k = \{i : p_{ki} \in C \vee q_{ki} \in C \vee r_{ki} \in C\}$. We have $\sum_k |X_k| \leq |C|$ and thus the smallest of X_k contains at most $|C|/|V'|$ elements. Each X_k is a dominating set in G': to distinguish $\{p_{ki}, q_{ki}\}$, there must be a beacon in p_{ki}, in q_{ki}, or in a neighbouring r_{kj}.

Assume that we can approximate 1-IC or 1-LDC within $1 + \alpha \ln |V|$ for some $\alpha > 0$. Let X^* denote a minimum dominating set in G'. There is a 1-IC and 1-LDC for G of size $|X^*||V'|+4|V'|$. The approximation algorithm returns a code of size at most $(1+\alpha \ln |V|)(|X^*||V'|+4|V'|)$. Use the code to construct a dominating set of size at most $(1+\alpha \ln |V|)(|X^*||V'|+4|V'|)/|V'| = (1+\alpha \ln(3|V'|^2 + 4|V'|))(|X^*|+4)$, which is less than $\gamma(1+\alpha\beta \ln |V'|)|X^*|$ for some positive constants β and γ. If $\ln |V'| \leq 1/\alpha\beta$ we can find a minimum dominating set in constant time by exhaustive search. Otherwise, $\gamma(1 + \alpha\beta \ln |V'|) \leq 2\alpha\beta\gamma \ln |V'|$. We have proved the following theorem.

Theorem 1 *Both* t-IC *and* t-LDC *are approximable within* $O(\log |V|)$ *for all* t*. However, there is a constant* $\rho > 0$ *such that for any constant* $\alpha > 0$*, a polynomial-time* $(1 + \alpha \ln |V|)$*-approximation algorithm for* 1-IC *or* 1-LDC *implies a polynomial-time* $\max\{1, \rho\alpha \ln |V|\}$*-approximation algorithm for* DOMINATING SET*.*

For the hardness of approximating DOMINATING SET within a logarithmic factor, see, for example, Lund and Yannakakis [17].

3 Approximability in Bounded-Degree Graphs

If the degree of the input graph is bounded by a constant, also $|S(c)|$ is bounded by a constant. As a t-IC or t-LDC has to cover all vertices, we need $\Omega(|V|)$ beacons. Thus, the trivial solution $C = V$ is a constant-factor approximation algorithm for t-IC and t-LDC in graphs of bounded degree. We prove that this result is asymptotically tight for $t = 1$, i.e., we cannot make the constant factor arbitrarily small.

Let us assume that for each constant Δ, there is a PTAS for 1-IC or 1-LDC in graphs of maximum degree Δ. We show how these schemes can be used to approximate Dominating Set within $1+\epsilon'$ in graphs of maximum degree Δ' for any constants $\epsilon' > 0$ and Δ'. Choose a constant $\epsilon > 0$ such that $(1+\epsilon)^2 < 1+\epsilon'$ and a positive integer μ such that $4(\Delta'+1)/(\mu-1) \le \epsilon$. Let $A = \{1, 2, \ldots, \mu\}$.

Let $G' = (V', E')$ be an instance of bounded-degree Dominating Set. Construct a graph $G = (V, E)$ as follows. The set of vertices V consists of p_{ki}, q_{ki}, r_{ki}, ϕ_{ai}, and ψ_{ai} for all $k \in A \diamond A$, $a \in A$, and $i \in V'$. The set of edges E consists of $\{p_{ki}, \phi_{ai}\}$, $\{p_{ki}, \phi_{bi}\}$, $\{q_{ki}, \phi_{ai}\}$, $\{q_{ki}, \phi_{bi}\}$, $\{r_{ki}, \psi_{ai}\}$, and $\{r_{ki}, \psi_{bi}\}$ for all $k = \{a, b\}$, $i \in V'$; and $\{p_{ki}, r_{kj}\}$ for all i, j, k such that either $i = j$ or $\{i, j\} \in E'$. See Fig. 2 for an illustration.

The maximum degree of G is bounded by the constant $\Delta = \max\{\Delta'+3, 2\mu-2\}$: The degree of each ψ_{ai} equals $\mu - 1$, as there are $\mu - 1$ distinct elements $b \in A$ such that $k = \{a, b\} \in A \diamond A$. Similarly, the degree of each ϕ_{ai} equals $2\mu - 2$. The degree of each q_{ki} equals 2, because the only neighbours are ϕ_{ai} and ϕ_{bi} for $k = \{a, b\}$. The degree of each p_{ki} is at most $\Delta'+3$, because the neighbours are the vertices ϕ_{ai} and ϕ_{bi} for $k = \{a, b\}$, the vertex r_{ki}, and at most Δ' vertices r_{kj} with $\{i, j\} \in E'$. The case of r_{ki} is analogous to p_{ki}.

As in Section 2, a dominating set X in G' can be used to construct a code $C = \{r_{ki} : k \in A \diamond A, i \in X\} \cup \{\phi_{ai}, \psi_{ai} : a \in A, i \in V'\}$ for G of size $|X||A \diamond A| + 2|V'||A|$, and a code C for G can be used to construct $|A \diamond A|$ dominating sets $X_k = \{i : p_{ki} \in C \vee q_{ki} \in C \vee r_{ki} \in C\}$ in G' of total size at most $|C|$.

Use the PTAS to approximate 1-IC or 1-LDC within $1+\epsilon$ in the constructed bounded-degree graph. Let X^* denote a minimum dominating set in G'. As each vertex dominates at most Δ' other vertices, we have $(\Delta'+1)|X^*| \ge |V'|$. The approximation algorithm returns a code of size at most $(1+\epsilon)(|X^*||A \diamond A| + 2|V'||A|)$, and we can use the code to construct a dominating set of size at most $(1+\epsilon)(|X^*||A \diamond A| + 2|V'||A|)/|A \diamond A| = (1+\epsilon)(|X^*| + 4|V'|/(\mu-1)) \le (1+\epsilon)(|X^*| + 4(\Delta'+1)|X^*|/(\mu-1)) \le (1+\epsilon')|X^*|$. We have proved the following theorem.

Theorem 2 *Both* t-IC *and* t-LDC *are approximable within a constant factor in graphs of bounded degree. However, if there is a PTAS for* 1-IC *or* 1-LDC *in graphs of bounded degree, there is a PTAS for* Dominating Set *in graphs of bounded degree.*

Dominating Set in graphs of bounded degree is APX-complete [11,19].

4 Approximability in Local Graphs

To find realistic problem classes that do admit a PTAS, we study the following family of graphs. We say that a graph is (d, N)-*local* if each vertex is associated with a point in $\mathbb{R}^d$ so that within any ball of radius 1, there are at most N vertices; and for each edge, the distance between the vertices is at most 1. Our definition of local graphs is similar in nature to *civilised graphs*, i.e., *graphs drawn in a civilised manner* [6, §8.5].

With suitable scaling of the space, the family of local graphs captures the features of many proposed applications of identifying codes. Consider, for example, a motion-detecting application: vertices correspond to physical areas, not arbitrarily small, and the length of each edge is limited by the maximum range of the sensor.

Local graphs are bounded-degree graphs but not necessarily graphs of a constant treewidth (consider a grid graph). In this section, we consider t-IC and t-LDC in the family of (d, N)-local graphs for a constant d and a constant N.

Fix the parameters d and N. Choose any $\epsilon > 0$. We show how to approximate t-IC and t-LDC within $1+\epsilon$ if the graph is (d, N)-local, by applying a shifting strategy; cf., e.g., Hunt et al. [8]. Choose an integer $m > 2^d d/\epsilon$. Consider all functions $f\colon \{1, 2, \ldots, d\} \to \{0, 1, \ldots, m-1\}$ and $g\colon \{1, 2, \ldots, d\} \to \mathbb{Z}$. Form a family of hypercubes $Q(f, g, r) = \{x \in \mathbb{R}^d : -r \le x_k/t - 4(mg(k) + f(k)) < 4(m-1) + r \ \forall k\}$. Intuitively, f selects one of m^d positions for a modular grid, g selects one cube in the grid, and r is the width of a margin around each cube; see Fig. 3. Denote by $V(f, g, r)$ the set of all vertices that are contained in $Q(f, g, r)$. For each pair (f, g) with a non-empty $V(f, g, 4)$, use exhaustive search to find the smallest set $C(f, g) \subseteq V(f, g, 4)$ that covers all vertices in $V(f, g, 3)$ and distinguishes all pairs in $V(f, g, 3) \diamond V(f, g, 3)$. Let $C(f) = \bigcup_g C(f, g)$. Choose a function f^* that minimises $|C(f^*)|$ and let $C = C(f^*)$.

Let us now prove the correctness of this algorithm. Let C^* be a minimum code. First, we show that for all (f, g), there is a $C(f, g)$ that satisfies the above conditions, and $|C(f, g)| \le |C^* \cap V(f, g, 4)|$. For each vertex $v \in V(f, g, 3)$, there is a beacon $c \in C^*$ such that c covers v. The distance from c to v is at most t, implying that $c \in V(f, g, 4)$. Similarly, for each pair $\{u, v\} \in V(f, g, 3) \diamond V(f, g, 3)$, there is a beacon $c \in C^*$ such that c distinguishes $\{u, v\}$. This is not possible if the distance from c to both u and v is more than t; thus, $c \in V(f, g, 4)$. It follows that the set $C^* \cap V(f, g, 4)$ satisfies the conditions.

Second, we show that the set C is a code. Consider any $v \in V$. For each f, there is at least one g such that $v \in V(f, g, 3)$. Thus, v is covered by $C(f^*, g)$ and by $C = C(f^*)$. Consider any pair $\{u, v\} \in V \diamond V$. If there is a g such that $u, v \in V(f^*, g, 3)$, the pair is distinguished by construction. If no such g exists, the distance between u and v is more than $2t$ units, as neighbouring hypercubes $Q(f^*, \cdot, 3)$ overlap by $2t$ units. Consider any beacon $c \in C$ that covers the vertex u. The distance from c to u is at most t units. This implies that the distance from c to v is more than t units. Thus $h_{uc} \neq 0$ and $h_{vc} = 0$, and c distinguishes the pair $\{u, v\}$.

Third, we show that $|C| \leq (1+\epsilon)|C^*|$. For each k, let $P_k(i) = \{x \in \mathbb{R}^d : -4 \leq x_k/t - 4(mj+i) < 0, j \in \mathbb{Z}\}$. Denote by $U_k(i)$ the set of vertices contained in $P_k(i)$, and let $U(f) = \bigcup_k U_k(f(k))$. The sets $P_k(\cdot)$ partition the space into m parts; thus, there is a function f' such that $|C^* \cap U_k(f'(k))| \leq |C^*|/m$ and $|C^* \cap U(f')| \leq d|C^*|/m$. Let $W(f,g) = V(f,g,4) \setminus V(f,g,0) \subseteq U(f)$. For each (f, v), there are at most 2^d functions g such that $v \in W(f,g)$. We get $|C| \leq |C(f')| = |\bigcup_g C(f',g)| \leq \sum_g |C(f',g)| \leq \sum_g |C^* \cap V(f',g,4)| = \sum_g |C^* \cap V(f',g,0)| + \sum_g |C^* \cap W(f',g)| \leq |C^*| + 2^d|C^* \cap U(f')| \leq (1 + 2^d d/m)|C^*| \leq (1+\epsilon)|C^*|$.

5 Fractional Packing of Codes

So far, we have focused on finding one code. This is relevant in deployment planning: given a graph that describes the landscape, decide where to place the beacons. Alternatively, we may be interested in operation planning: given a deployed system that consists of battery-powered devices (e.g., a wireless sensor network [13]), decide how to schedule the activity of the devices to maximise the lifetime of the system. More precisely, we want to find a *sleep schedule*: during each time interval, the set of active devices forms an identifying or locating-dominating code. The time that each device can act as a beacon is bounded by its battery capacity.

This leads into a fractional packing problem: maximise $\sum x_j$ subject to $Ax \leq b$ and $x \geq 0$. The columns of the matrix $A = (a_{ij})$ consist of 0-1 vectors that describe all possible codes, b_i is the battery capacity of the ith vertex, and x_j is the time interval allocated for the jth code. Note that a collection of *disjoint* identifying codes [14] provides a feasible but not necessarily optimal solution.

To solve this LP, we may apply, for example, the approximation scheme by Garg and Könemann [7]. In the scheme, we need to provide an oracle that finds a minimum-weight column of A: given a nonnegative weight vector w, the oracle has to find a column j that minimises $\sum_i w_i a_{ij}$. With an exact oracle we obtain a PTAS for fractional packing, but we may also use an approximate oracle, obtaining an approximation algorithm for fractional packing.

This raises the issue of the computational complexity of WEIGHTED t-IC and WEIGHTED t-LDC. In the case of general graphs, we can translate the problem into an instance of WEIGHTED SET COVERING; again, we obtain a logarithmic approximation ratio [3]. In local graphs, we can use the same approach as in Section 4. Our polynomial-time approximation scheme is designed so that it directly generalises to weighted problems; the only difference is that the cardinalities of the sets are replaced by sums of weights. The approximability of WEIGHTED t-IC and WEIGHTED t-LDC in bounded-degree graphs remains an open problem.

Acknowledgements. Thanks to Patrik Floréen, Petteri Kaski, Jukka Kohonen, and Tanja Säily for comments and discussions, and to an anonymous referee for valuable suggestions.

This work was supported in part by the Academy of Finland, Grant 116547, by Helsinki Graduate School in Computer Science and Engineering (Hecse), and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the author's views.

Notes added after preparing the final version. During the review of this paper, similar results have been published independently in the following work: M. Laifenfeld, A. Trachtenberg, and T.Y. Berger-Wolf. Identifying codes and the set cover problem. In *Proc. 44th Annual Allerton Conference on Communication, Control, and Computing (Monticello, IL, USA, September 2006).*

References

[1] I. Charon, O. Hudry, and A. Lobstein. Identifying and locating-dominating codes: NP-completeness results for directed graphs. *IEEE Transactions on Information Theory*, 48(8):2192–2200, 2002.

[2] I. Charon, O. Hudry, and A. Lobstein. Minimizing the size of an identifying or locating-dominating code in a graph is NP-hard. *Theoretical Computer Science*, 290(3):2109–2120, 2003.

[3] V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[4] G. Cohen, I. Honkala, A. Lobstein, and G. Zémor. On identifying codes. In *Proc. DIMACS Workshop on Codes and Association Schemes (Piscataway, NJ, USA, 1999)*, volume 56 of *DIMACS Series in Discrete Mathematics*

and Theoretical Computer Science, pages 97–109, Providence, RI, USA, 2001. American Mathematical Society.

[5] C. J. Colbourn, P. J. Slater, and L. K. Stewart. Locating dominating sets in series parallel networks. *Congressus Numerantium*, 56:135–162, 1987.

[6] P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. Number 22 in The Carus Mathematical Monographs. The Mathematical Association of America, Washington, DC, USA, 1984.

[7] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proc. 39th Annual Symposium on Foundations of Computer Science (FOCS, Palo Alto, CA, USA, November 1998)*, pages 300–309, Los Alamitos, CA, USA, 1998. IEEE Computer Society.

[8] 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*, 26(2):238–274, 1998.

[9] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[10] M. Karpovsky, K. Chakrabarty, and L. Levitin. On a new class of codes for identifying vertices in graphs. *IEEE Transactions on Information Theory*, 44(2):599–611, 1998.

[11] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *SIAM Journal on Computing*, 28(1):164–191, 1999.

[12] S. Khuller, B. Raghavachari, and A. Rosenfeld. Landmarks in graphs. *Discrete Applied Mathematics*, 70(3):217–229, 1996.

[13] B. Krishnamachari. *Networking Wireless Sensors*. Cambridge University Press, Cambridge, UK, 2005.

[14] M. Laifenfeld and A. Trachtenberg. Disjoint identifying-codes for arbitrary graphs. In *Proc. International Symposium on Information Theory (ISIT, Adelaide, Australia, September 2005)*, pages 244–248, Piscataway, NJ, USA, 2005. IEEE.

[15] K. B. Lakshmanan, D. J. Rosenkrantz, and S. S. Ravi. Alarm placement in systems with fault propagation. *Theoretical Computer Science*, 243(1–2):269–288, 2000.

[16] A. Lobstein. Codes identifiants et localisateurs-dominateurs dans les graphes: une bibliographie, January 2007. `http://perso.enst.fr/~lobstein/bibLOCDOMetID.html`

[17] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

[18] J. Moncel. Optimal graphs for identification of vertices in networks. Technical Report 138, Laboratoire Leibniz, Grenoble, France, November 2005.

[19] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[20] P. J. Slater. Domination and location in acyclic graphs. *Networks*, 17(1):55–64, 1987.

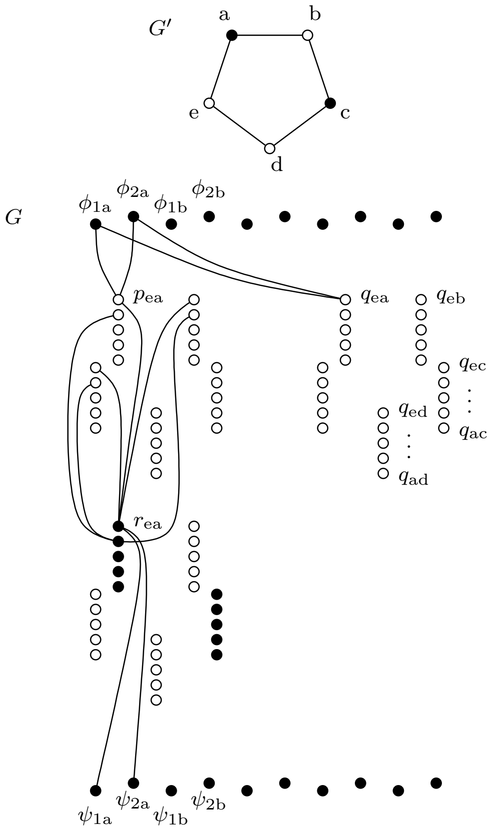

Fig. 1. The reduction for general graphs. In this example, we have chosen $f(\mathrm{e}, \mathrm{a}) = \{(1, \mathrm{a}), (2, \mathrm{a})\}$. The black vertices in G' are a dominating set, and the black vertices in G are a 1-IC and a 1-LDC. All edges from r_{ea} are shown, as well as selected other edges.

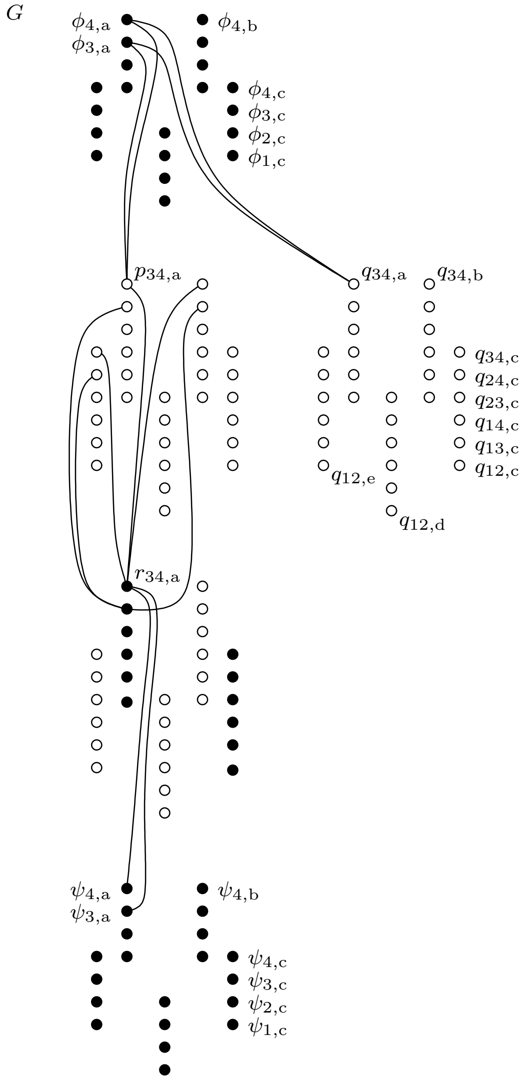

Fig. 2. The reduction for bounded-degree graphs. In this example, $\mu = 4$. To simplify the illustration, we write $A \diamond A = \{12, 13, 14, 23, 24, 34\}$ instead of $\{\{1, 2\}, \{1, 3\}, \dots\}$. The original graph G' is the same as in Fig. 1. The black vertices in G are a 1-IC and a 1-LDC.

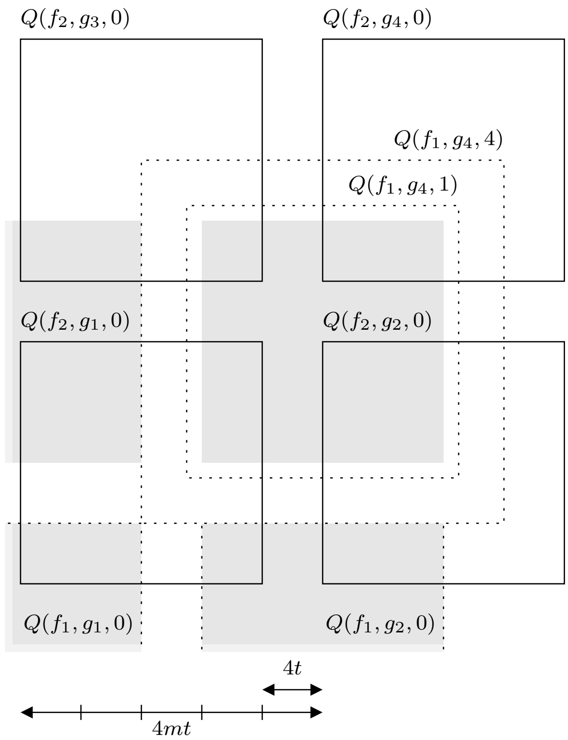

Fig. 3. The modular grid in the 2-dimensional case. Here $m = 5$, $f_1 = (0, 0)$, $f_2 = (2, 3)$, $g_1 = (1, 1)$, $g_2 = (2, 1)$, $g_3 = (1, 2)$, and $g_4 = (2, 2)$.