Available online at www.sciencedirect.com

ScienceDirect

Information Processing Letters 101 (2007) 101–106

Information Processing Letters

www.elsevier.com/locate/ipl

An improved exact algorithm for the domatic number problem ☆

Tobias Riege [a], Jörg Rothe [a,*,1], Holger Spakowski [a], Masaki Yamamoto [b,2]

[a] *Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany*
[b] *Department of Mathematical and Computing Science, Tokyo Institute of Technology, Tokyo, Japan*

Received 20 March 2006; received in revised form 12 August 2006; accepted 21 August 2006

Available online 5 October 2006

Communicated by K. Iwama

Abstract

The 3-domatic number problem asks whether a given graph can be partitioned into three dominating sets. We prove that this problem can be solved by a deterministic algorithm in time 2.695^n (up to polynomial factors) and in polynomial space. This result improves the previous bound of 2.8805^n, which is due to Björklund and Husfeldt. To prove our result, we combine an algorithm by Fomin et al. with Yamamoto's algorithm for the satisfiability problem. In addition, we show that the 3-domatic number problem can be solved for graphs G with bounded maximum degree $\Delta(G)$ by a randomized polynomial-space algorithm, whose running time is better than the previous bound due to Riege and Rothe whenever $\Delta(G) \geqslant 5$. Our new randomized algorithm employs Schöning's approach to constraint satisfaction problems.

Keywords: Graph algorithms; Exact algorithms; Randomized algorithms; Domatic number problem

☆ Supported in part by the German Science Foundation under grants RO 1202/9-1 and RO 1202/9-3. A preliminary version of this work was presented at ICTTA'06 (see [T. Riege, J. Rothe, H. Spakowski, M. Yamamoto, An improved exact algorithm for the domatic number problem, in: Proceedings of the Second IEEE International Conference on Information & Communication Technologies: From Theory to Applications, IEEE Computer Society Press, April 2006, pp. 1021–1022], which improves on the results of [T. Riege, J. Rothe, An exact 2.9416^n algorithm for the three domatic number problem, in: Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science, in: Lecture Notes in Computer Science, vol. 3618, Springer-Verlag, Berlin, August 2005, pp. 733–744]) and also at the ICALP-affiliated workshop iETA'06.
* Corresponding author.
E-mail addresses: riege@cs.uni-duesseldorf.de (T. Riege), rothe@cs.uni-duesseldorf.de (J. Rothe), spakowsk@cs.uni-duesseldorf.de (H. Spakowski), masaki.yamamoto@lab2.kuis.kyoto-u.ac.jp (M. Yamamoto).
[1] Supported in part by the Alexander von Humboldt Foundation in the CONNECT program. Work done in part while visiting the Tokyo Institute of Technology.
[2] Current affiliation: Department of Communications and Computer Engineering, Kyoto University, Kyoto, Japan.

0020-0190/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2006.08.010

1. Introduction

A dominating set in an (undirected) graph G is a subset D of the vertex set V of G such that the closed neighborhood of D equals V. The domatic number problem asks for a partition of V into a maximum number of disjoint dominating sets. This number, denoted by $\delta(G)$, is called the domatic number of G. The domatic number problem arises in the area of computer networks and is related to the tasks of distributing resources and of locating facilities in the network. This problem has been intensely studied, see, e.g., [3,4,2,13,11,7,8,16].

For each $k \geqslant 3$, it is NP-complete to determine whether or not the domatic number of a given graph is at least k, see Garey and Johnson [9]. That is why we cannot expect to find a polynomial-time algorithm that solves the problem.

Among the various ways of coping with NP-hard problems (such as approximation, randomization, or parameterized complexity), much attention has been paid to designing exact exponential-time algorithms for such problems that are better than the trivial exponential-time algorithm. In particular, if the trivial algorithm runs in time 3^n but one is able to find a c^n algorithm for this problem with $c < 3$ (up to polynomial factors), then this algorithm can handle larger problem instances in the same amount of time than the trivial algorithm. This difference can be quite important in practice. For recent surveys on this subject, we refer to Schöning [19] and Woeginger [20].

The first result breaking the trivial 3^n barrier for the 3-domatic number problem is due to Riege and Rothe [17], who proposed a polynomial-space algorithm running in time 2.9416^n. Fomin et al. [6] gave an exponential-space, 2.8805^n-time algorithm. Very recently, Björklund and Husfeldt [1] improved their result by providing a 2^n-time bound for several partition problems including the domatic number problem. Their exact algorithm is based on a novel inclusion–exclusion formula and works in exponential space. They also proposed a polynomial-space algorithm for computing the domatic number, which runs in time 2.8805^n.

In this paper, we obtain a further improvement by making use of two known algorithms: an algorithm by Fomin et al. [6] for generating all minimal dominating sets of a graph, and Yamamoto's algorithm for the satisfiability problem [21]. We show that these two algorithms can be combined so as to yield a 2.695^n-time algorithm for the 3-domatic number problem that runs in *polynomial space*.

In addition, we prove that there is a randomized polynomial-space algorithm solving this problem for graphs with bounded maximum degree, which improves the previous bound due to Riege and Rothe [17]. Here, we apply Schöning's results for constraint satisfaction problems [18], which previously was also useful in improving the bounds of randomized exponential-time algorithms for the satisfiability problem. For example, Iwama and Tamaki [12] designed a randomized algorithm with running time 1.324^n for solving `3-SAT` by combining Schöning's algorithm with an algorithm due to Paturi et al. [15].

2. Preliminaries

We first introduce some graph-theoretical notation. Graphs are pairs $G = (V, E)$, where V is the vertex set of G and E is the edge set of G. All graphs considered in this paper are undirected and simple. That is, the edges of any graph are unordered pairs of vertices, and there are neither multiple nor reflexive edges; i.e., there exists at most one edge of the form $\{x, y\}$ for any two vertices x and y, and there is no edge of the form $\{x, x\}$. Moreover, we require all graphs to not have isolated vertices. In general, graphs need not be connected.

Definition 1. Let $G = (V, E)$ be a graph.

- For any vertex $v \in V$, define the *neighborhood of v in G* by

  $N(v) = \{u \in V \mid \{u, v\} \in E\}$,

  and define the *closed neighborhood of v in G* by

  $N[v] = \{v\} \cup N(v)$.
- For any subset $U \subseteq V$ of the vertices of G, define $N[U] = \bigcup_{u \in U} N[u]$ and $N(U) = N[U] - U$.
- A subset $D \subseteq V$ is a *dominating set of G* if and only if every vertex $u \notin D$ is a neighbor of some vertex $v \in D$. That is, D dominates G if and only if $N[D] = V$.
- A dominating set D is called a *minimal dominating set* if and only if there exists no dominating set C of G with $C \subset D$.
- The *domatic number of G* (denoted by $\delta(G)$) is the maximum number of disjoint dominating sets.
- Given a graph G and a positive integer k, the *k-domatic number problem*, `k-DNP`, for short, asks whether or not $\delta(G) \geqslant k$.

Note that at least one partition of G into $\delta(G)$ (i.e., the maximum number of) disjoint dominating sets contains $\delta(G) - 1$ minimal dominating sets. To see why, fix in a given partition some dominating set of G, say D_0, and note that for each of the remaining $\delta(G) - 1$ dominating sets (call any such set D), we can move a vertex v from D to D_0 until $D - \{v\}$ becomes minimal.

As noted earlier, for $k \geqslant 3$, the k-domatic number problem is NP-complete, see Garey and Johnson [9]. Therefore, no polynomial-time algorithm for k-DNP exists unless $\mathrm{P} = \mathrm{NP}$.

Here, we focus on the case $k = 3$ only. The first algorithm beating the trivial $\widetilde{\mathcal{O}}(3^n)$ barrier[3] for the 3-domatic number problem is due to Riege and Rothe [17]. They also investigate this problem for graphs with bounded maximum degree, and they propose a deterministic and a randomized algorithm for it. In particular, this deterministic algorithm outperforms their general deterministic algorithm for the 3-domatic number problem whenever $\Delta(G)$, the maximum degree of the input graph G, is in the range $3 \leqslant \Delta(G) < 7$.

Fomin et al. [6] recently improved the general result from [17] by constructing an algorithm for the domatic number problem with running time $\widetilde{\mathcal{O}}(2.8805^n)$. Their algorithm makes use of a new algorithm for enumerating all minimal dominating sets of a graph.

Theorem 2. (Fomin et al. [6].) *There is an algorithm for listing all minimal dominating sets in an n vertex graph G in time $\widetilde{\mathcal{O}}(1.7697^n)$ and in polynomial space.*

Their proof relies on a new method to evaluate the size of the recursion tree for a given exponential-time algorithm. This technique, which is called "measure and conquer", was introduced by Fomin et al. [5], who applied it to give a better analysis of the runtimes of exact backtracking algorithms for the minimum dominating set problem and the minimum set cover problem.

The "measure and conquer" technique is based on the following idea. By choosing a suitable measure of the subproblems generated by the recursive algorithm considered, one can lower-bound the progress made by the algorithm in each branching step. A clever choice of this measure can yield a much better worst-case runtime analysis of the problem, even though the algorithm considered is not new and may have long been known.

Based on this new technique, Fomin et al. [6] design an exponential-time algorithm for determining the domatic number of a given graph (and thus, in particular, for solving 3-DNP). Their approach resembles the dynamic-programming algorithm by Lawler [14] for computing the chromatic number of a graph.

Theorem 3. (Fomin et al. [6].) *There is an algorithm for computing the domatic number of a given graph that runs in time $\widetilde{\mathcal{O}}(2.8805^n)$ (and in exponential space).*

Björklund and Husfeldt [1] very recently proposed an exponential-space, $\widetilde{\mathcal{O}}(2^n)$-time exact algorithm for several partition problems including the domatic number problem. They also proposed a polynomial-space algorithm for this problem.

Theorem 4. (Björklund and Husfeldt [1].) *There is an exponential-space algorithm for computing the domatic number of a given graph that runs in time $\widetilde{\mathcal{O}}(2^n)$. There is also a polynomial-space algorithm for this problem that runs in time $\widetilde{\mathcal{O}}(2.8805^n)$.*

Prior to this paper, the time bound stated in Theorem 4 was the best result known for the (search version of the) domatic number problem, and in particular for 3-DNP. In the next section, combining the method of [6] with the SAT algorithm of [21], we improve the time bound of the polynomial-space algorithm of Björklund and Husfeldt [1] for the decision problem 3-DNP.

3. An improved exact algorithm for the 3-domatic number problem

In this section, we improve the exponential running time for 3-DNP that follows from Theorem 4 for the case of polynomial-space bounded computation. To this end, we combine the algorithm for enumerating all minimal dominating sets from Theorem 2 with an algorithm due to Yamamoto [21].

Yamamoto's algorithm, which is based on (and improves) an algorithm due to Hirsch [10], solves the NP-complete problem SAT in time $\widetilde{\mathcal{O}}(1.234^m)$, where m is the number of clauses of the given boolean formula in conjunctive normal form. To prove Theorem 6 below, we will apply Yamamoto's algorithm to a special version of SAT, which is called NAE-SAT ("not-all-equal satisfiability").

Definition 5. Let $\varphi = \varphi(X, C)$ be a boolean formula in conjunctive normal form consisting of a collection $C = \{c_1, c_2, \ldots, c_m\}$ of m clauses over the variable set X. We say φ is in NAE-SAT if and only if there exists a truth assignment for X satisfying all clauses in C and such that in none of the clauses, all literals are true.

[3] As is common for exponential-time algorithms, we use the $\widetilde{\mathcal{O}}$ notation to indicate that polynomial factors are neglected. That is, for functions f and g, we write $f \in \widetilde{\mathcal{O}}(g)$ if $f \in \mathcal{O}(p \cdot g)$, where p is some polynomial.

Theorem 6. *There exists an exact polynomial-space algorithm solving the* 3-DNP *problem in time* $\widetilde{\mathcal{O}}(2.695^n)$.

Proof. Let $G=(V,E)$ be a given graph with n vertices. Using the algorithm from Theorem 2, we generate all minimal dominating sets of G sequentially, where we are not "enumerating" them all but rather are re-using the same space for each iteration to make sure that our algorithm runs in polynomial space. Each time a minimal dominating set $D\subseteq V$ of G is generated, we create a formula $\varphi_D=\varphi_D(X,C)$ for the NP-complete problem NAE-SAT as follows:

- The set of variables is defined as

  $X=\{x_v \mid v\in V-D\}$.
- For each vertex $v\in V$, create the clause

  $$C_v=\Big\{\bigcup_{\substack{u\in N[v]\\ u\notin D}} x_u\Big\},$$

  so the clause set is defined as $C=\{C_v \mid v\in V\}$.

If $G\in$ 3-DNP then one of the partitions into three dominating sets of G contains a minimal dominating set D, see the comment right after Definition 1. Let the other two dominating sets be D_0 and D_1. Every vertex $v\in V$ is in $N[D_0]$ and also in $N[D_1]$. It follows that $\varphi_D\in$ NAE-SAT, via the truth assignment that maps every variable corresponding to a vertex in D_0 to 0 (representing *false*) and every variable corresponding to a vertex in D_1 to 1 (representing *true*).

Conversely, if $\varphi_D\in$ NAE-SAT for some minimal dominating set D, construct two additional dominating sets, D_0 and D_1, for G as follows. Let t be a satisfying truth assignment for φ_D in the not-all-equal sense. Set $D_0=\{v\in V-D \mid t(x_v)=0\}$ and $D_1=\{v\in V-D \mid t(x_v)=1\}$. Since for every vertex $v\in V$ clause C_v contains both true and false literals, D_0 and D_1 are dominating sets of G.

It follows that $G\in$ 3-DNP if and only if φ_D is in NAE-SAT for some formula φ_D thus defined.

The number of clauses in φ_D equals n, the number of vertices in G. We use the following easy, well-known reduction from NAE-SAT to SAT. For each clause $C_v=(\ell_1\vee\cdots\vee\ell_k)$, we add the clause $C_{\bar v}=(\overline{\ell_1}\vee\cdots\vee\overline{\ell_k})$ to the clause set, where $\overline{\ell_i}$ denotes the negation of literal ℓ_i. We obtain a formula $\varphi'_D=\varphi'_D(X,C\cup C^-)$, where $C^-=\{C_{\bar v} \mid v\in V\}$, with the property that

$$\varphi_D\in \text{NAE-SAT} \iff \varphi'_D\in \text{SAT}.$$

Using the exponential-time SAT algorithm designed by Yamamoto [21], we can now determine the satisfiability of φ'_D. Note that φ'_D has $2n$ clauses, so Yamamoto's algorithm runs in time $\widetilde{\mathcal{O}}(1.234^{2n})$. It follows that our algorithm to solve 3-DNP has a running time of

$$\widetilde{\mathcal{O}}\big(1.7697^n\cdot 1.234^{2n}\big)=\widetilde{\mathcal{O}}\big(2.695^n\big).$$

It runs in polynomial space, since it takes polynomial space to enumerate all minimal dominating sets and to execute the above SAT algorithm. □

4. An improved randomized algorithm for the 3-domatic number problem

We now turn to the case where the maximum degree $\Delta(G)$ of the input graph G is bounded by some small constant. Riege and Rothe [17] described a randomized algorithm for this problem. Here, we observe that their result can be improved by using Schöning's algorithm for constraint satisfaction problems [18].

Theorem 7. *There is a randomized polynomial-space algorithm solving the* 3-DNP *problem for graphs G with bounded maximum degree* $\Delta(G)$, *whose running time depends on* $\Delta(G)$ *as stated in Table* 1.

Proof. The 3-DNP problem for a graph G can easily be formulated as a constraint satisfaction problem (CSP) on the domain $D=\{0,1,2\}$ with each constraint having order $\ell\leqslant\Delta(G)+1$. The order of a constraint is the number of arguments in the constraint.

Let $G=(V,E)$ be a given graph. Create the constraint satisfaction problem F with n variables as follows:

- The set of variables is defined as

  $X=\{x_v \mid v\in V\}$.
- For each vertex $v\in V$, create the constraint C_v defined by

  $C_v(x_v,x_{w_1},x_{w_2},\ldots,x_{w_{\|N(v)\|}})=1 \iff$

  all values of $D=\{0,1,2\}$ appear in

  the values of $x_v,x_{w_1},x_{w_2},\ldots,x_{w_{\|N(v)\|}}$,

  where $w_1,w_2,\ldots,w_{\|N(v)\|}$ are the vertices adjacent to v.

It is easy to see that

$$G\in \text{3-DNP} \iff \text{CSP } F \text{ has a solution.}$$

Now we can apply Schöning's randomized (polynomial-space) algorithm for solving constraint satisfaction problems [18] to determine whether F has a solution.

Table 1
Three 3-DNP algorithms for graphs G with bounded maximum degree $\Delta(G)$

$\Delta(G)$	3	4	5	6	7	8	Source
Deterministic	2.2894^n	2.6591^n	2.8252^n	2.9058^n	2.9473^n	2.9697^n	[17]
Randomized	2^n	2.3570^n	2.5820^n	2.7262^n	2.8197^n	2.8808^n	[17]
Randomized	2.2501^n	2.4001^n	2.5001^n	2.5715^n	2.6251^n	2.6667^n	Thm. 7

We thus obtain a randomized polynomial-space algorithm for 3-DNP with running time

$$\widetilde{\mathcal{O}}\left(3\left(1-\frac{1}{\Delta(G)+1}\right)+\varepsilon\right)^n,$$

for any $\varepsilon > 0$. □

For graphs with $\Delta(G) \geqslant 5$, this is an improvement over the randomized algorithm given by Riege and Rothe [17], see Table 1.

5. Conclusions

In this paper, we considered the 3-domatic number problem, which asks whether a given graph can be partitioned into three dominating sets, and we have shown how to improve on existing exact and randomized exponential-time algorithms for this problem.

In particular, by reducing 3-DNP to the NAE-SAT problem and by combining Yamamoto's algorithm [21] with the algorithm by Fomin et al. [6], we obtained an exact (i.e., deterministic) algorithm that runs in time $\widetilde{\mathcal{O}}(2.695^n)$ and polynomial space. This result improves on the previously best bound of $\widetilde{\mathcal{O}}(2.8805^n)$ for the 3-domatic number problem, which was achieved by Björklund and Husfeldt [1]. An even earlier—and to our knowledge, the first—nontrivial bound for this problem is due to Riege and Rothe [17] who presented an $\widetilde{\mathcal{O}}(2.9416^n)$ algorithm.

We also described a randomized polynomial-space algorithm solving the 3-domatic number problem for graphs G with bounded maximum degree $\Delta(G)$. Whenever $\Delta(G) \geqslant 5$, the running time of our new randomized algorithm is better than the previously known bound, which is due to Riege and Rothe [17]. These results are summarized in Table 1. Our new randomized algorithm makes use of Schöning's algorithm for constraint satisfaction problems [18].

Acknowledgements

We are grateful to Osamu Watanabe and Dieter Kratsch for inspiring discussions on the subject of this paper. In particular, we thank Dieter Kratsch for calling our attention to Lawler's algorithm. The second author thanks Osamu Watanabe for his kind hospitality when hosting a research visit at the Tokyo Institute of Technology in September and October 2005 that brought about this collaboration.

References

[1] A. Björklund, T. Husfeldt, Inclusion–exclusion algorithms for counting set partitions, in: Proceedings of the 47th IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, October 2006, in press.

[2] M. Bonuccelli, Dominating sets and dominating number of circular arc graphs, Discrete Applied Mathematics 12 (1985) 203–213.

[3] E. Cockayne, S. Hedetniemi, Towards a theory of domination in graphs, Networks 7 (1977) 247–261.

[4] M. Farber, Domination, independent domination, and duality in strongly chordal graphs, Discrete Applied Mathematics 7 (1984) 115–130.

[5] F. Fomin, F. Grandoni, D. Kratsch, Measure and conquer: Domination—A case study, in: Proceedings of the 32nd International Colloquium on Automata, Languages, and Programming, 2005, pp. 191–203.

[6] F. Fomin, F. Grandoni, A. Pyatkin, A. Stepanov, Bounding the number of minimal dominating sets: A measure and conquer approach, in: Proceedings of the 16th International Symposium on Algorithms and Computation, 2005, pp. 573–582.

[7] U. Feige, M. Halldórsson, G. Kortsarz, Approximating the domatic number, in: Proceedings of the 32nd ACM Symposium on Theory of Computing, ACM Press, May 2000, pp. 134–143.

[8] U. Feige, M. Halldórsson, G. Kortsarz, A. Srinivasan, Approximating the domatic number, SIAM Journal on Computing 32 (1) (2002) 172–195.

[9] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, 1979.

[10] E. Hirsch, New worst-case upper bounds for SAT, Journal of Automated Reasoning 24 (2000) 397–420.

[11] P. Heggernes, J. Telle, Partitioning graphs into generalized dominating sets, Nordic Journal of Computing 5 (2) (1998) 128–142.

[12] K. Iwama, S. Tamaki, Improved upper bounds for 3-SAT, in: J. Munro (Ed.), Proceedings of the 15th Annual ACM–SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, January 2004, pp. 328–329.

[13] H. Kaplan, R. Shamir, The domatic number problem on some perfect graph families, Information Processing Letters 49 (1) (1994) 51–56.

[14] E. Lawler, A note on the complexity of the chromatic number problem, Information Processing Letters 5 (3) (1976) 66–67.

[15] R. Paturi, P. Pudlák, M. Saks, F. Zane, An improved exponential-time algorithm for k-SAT, in: Proceedings of the 39th IEEE

Symposium on Foundations of Computer Science, IEEE Computer Society Press, November 1998, pp. 628–637.
[16] T. Riege, J. Rothe, Complexity of the exact domatic number problem and of the exact conveyor flow shop problem, Theory of Computing Systems 39 (5) (2006) 635–668.
[17] T. Riege, J. Rothe, An exact 2.9416^n algorithm for the three domatic number problem, in: Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science, in: Lecture Notes in Computer Science, vol. 3618, Springer-Verlag, Berlin, August 2005, pp. 733–744.
[18] U. Schöning, A probabilistic algorithm for k-SAT and constraint satisfaction problems, in: Proceedings of the 40th IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, October 1999, pp. 410–414.
[19] U. Schöning, Algorithmics in exponential time, in: Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science, in: Lecture Notes in Computer Science, vol. 3404, Springer-Verlag, Berlin, 2005, pp. 36–43.
[20] G. Woeginger, Exact algorithms for NP-hard problems, in: M. Jünger, G. Reinelt, G. Rinaldi (Eds.), Combinatorical Optimization: "Eureka, you shrink!", in: Lecture Notes in Computer Science, vol. 2570, Springer-Verlag, Berlin, 2003, pp. 185–207.
[21] M. Yamamoto, An improved $\widetilde{\mathcal{O}}(1.234^m)$-time deterministic algorithm for SAT, in: Proceedings of the 16th International Symposium on Algorithms and Computation, in: Lecture Notes in Computer Science, vol. 3827, Springer-Verlag, Berlin, 2005, pp. 644–653.