ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Information Processing Letters 85 (2003) 1–6

Information Processing Letters

www.elsevier.com/locate/ipl

Monotone Boolean dualization is in co-NP[$\log^2 n$]

Dimitris J. Kavvadias [a], Elias C. Stavropoulos [b,*]

[a] *University of Patras, Department of Mathematics, GR-265 00 Patras, Greece*
[b] *University of Patras, Computer Engineering & Informatics Department, GR-265 00 Patras, Greece*

Received 6 March 2002; received in revised form 5 June 2002

Communicated by H. Ganzinger

Abstract

In 1996, Fredman and Khachiyan [J. Algorithms 21 (1996) 618–628] presented a remarkable algorithm for the problem of checking the duality of a pair of monotone Boolean expressions in disjunctive normal form. Their algorithm runs in $n^{o(\log n)}$ time, thus giving evidence that the problem lies in an intermediate class between P and co-NP. In this paper we show that a modified version of their algorithm requires deterministic polynomial time plus $O(\log^2 n)$ nondeterministic guesses, thus placing the problem in the class co-NP[$\log^2 n$]. Our nondeterministic version has also the advantage of having a simpler analysis than the deterministic one.

Keywords: Algorithms; Computational complexity; Monotone DNF duality; Transversal hypergraph

1. Introduction

Let $f(x) = f(x_1, \dots, x_N)$ and $g(x) = g(x_1, \dots, x_N)$ be a pair of monotone Boolean expressions given by their irredundant disjunctive normal forms

$$f = \bigvee_{I \in F} \bigwedge_{i \in I} x_i \quad \text{and} \quad g = \bigvee_{J \in G} \bigwedge_{j \in J} x_j,$$

where F and G are the sets of prime implicants $I, J \subseteq \{1, \dots, N\}$ of f and g, respectively. The problem of interest here is defined as follows:

MONOTONE BOOLEAN DUALITY (or, MBD). Given a pair of monotone Boolean expressions f and g in their irredundant disjunctive normal forms, decide whether f, g are mutually dual, i.e.,

$$f(x_1, \dots, x_N) = \bar{g}(\bar{x}_1, \dots, \bar{x}_N) \quad \text{for all } x = (x_1, \dots, x_N) \in \{0, 1\}^N.$$

If f and g are not mutually dual, then there is a vector $x \in \{0, 1\}^N$ such that, $f(x_1, \dots, x_N) = g(\bar{x}_1, \dots, \bar{x}_N)$. Obviously, such a disqualifier can be guessed and verified in time that is polynomial to the size $n = |F| + |G|$ of f and g. Thus, MBD is easily placed in the class co-NP. However, its exact computational complexity is still unknown. The most notable result was given in 1996 by Fredman and Khachiyan [6]. In this work, the authors presented an algorithm that checks the duality of a pair of monotone DNFs in quasi-polynomial time $n^{o(\log n)}$. This result gives evidence that MBD lies in an intermediate class between P and co-NP.

---

[*] Corresponding author.

E-mail addresses: djk@math.upatras.gr (D.J. Kavvadias), estavrop@ceid.upatras.gr (E.C. Stavropoulos).

0020-0190/02/$ – see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0020-0190(02)00346-0

The algorithm of Fredman and Khachiyan can also be used for enumerating the prime implicants of the dual expression of a monotone DNF in *incremental output-subexponential* time [9]. Since the size of the dual expression may be exponentially larger than the input one, more elaborate complexity measures for the efficiency of algorithms for problems like MBD (i.e., with large output) must be defined, that will take into account not only the size of the input but the size of the output as well. The reader is referred to see [10,14] for discussions of algorithms with large output and performance criteria.

Generating the prime implicants of a monotone DNF is equivalent to the generation of the *minimal transversals* of a simple hypergraph [2,6,12] and to the generation of the *maximal models* of a Boolean expression in conjunctive normal form [11]. These problems are central in various fields of Computer Science (see [5,6,8] for an exposition of applications of these problems).

The algorithm of Fredman and Khachiyan gives an upper bound for the time complexity of MBD and implies that the problem can not be co-NP-hard, unless any co-NP-complete problem can be solved in quasi-polynomial time. In this paper we present a nondeterministic version of the algorithm of Fredman and Khachiyan. Our version uses the decomposition rules of [6] (see next section) in a novel way and solves the problem in deterministic polynomial time plus $O(\log^2 n)$ nondeterministic steps. Having the above time bounds, it is subsequently straightforward to obtain the $n^{o(\log n)}$ deterministic time bound of [6], thus avoiding the rather complicated analysis presented there. Hence, we place the MONOTONE BOOLEAN DUALITY problem in the class co-NP[$\log^2 n$], the subclass of co-NP where only the first $\log^2 n$ steps are nondeterministic. This is the complement of the class NP[$\log^2 n$], denoted β_2P in [13]. Such subclasses of NP can be defined by restricting the number of nondeterministic steps of the computation (see [13,3]). For a survey on limited nondeterminism, see [7]. The same complexity result was also given independently in [4]. Our approach differs from the one in [4] and its analysis is much simpler. Moreover, as it is mentioned in [4], the same result may also be obtained by appropriately applying Beigel and Fu's Theorem 11 in [1].

The rest of the paper is organized as follows: In Section 2 we present the necessary duality properties and lemmas and shortly describe the algorithm of Fredman and Khachiyan. In the next section we present our nondeterministic version and give its time complexity. Finally, in Section 4 some conclusions and directions for further research are given.

2. Overview of the Fredman and Khachiyan algorithm

For the sake of completeness, in this section we briefly describe the main steps of the Fredman and Khachiyan algorithm. The reader is referred to [6] for more details. Terminology and notation are also borrowed from there.

Suppose that the monotone DNFs f and g are mutually dual. Then, the following conditions hold:

$$I \cap J \neq \emptyset, \quad \text{for any } I \in F \text{ and } J \in G, \tag{1}$$

$$\bigcup \{I\colon I \in F\} = \bigcup \{J\colon J \in G\}, \tag{2}$$

$$\begin{aligned} &\max\bigl\{|I|\colon I \in F\bigr\} \leqslant |G|,\\ &\max\bigl\{|J|\colon J \in G\bigr\} \leqslant |F|. \end{aligned} \tag{3}$$

Moreover (cf. [6, Lemma 1]),

$$E = \sum_{I \in F} 2^{-|I|} + \sum_{J \in G} 2^{-|J|} \geqslant 1. \tag{4}$$

If any of these necessary duality conditions is violated, then f and g are not mutually dual and a succinct disqualifier can be found in polynomial time.

Let $x_i \in \{x_1, \ldots, x_N\}$. The *frequency* ε_i^f of x_i in f is defined as the fraction of implicants of F that x_i occurs in, i.e.,

$$\varepsilon_i^f = \frac{|\{I \in F\colon i \in I\}|}{|F|}.$$

Let $\varepsilon \in (0, 1]$. We say that x_i *occurs* in f with *frequency at least* ε if $\varepsilon_i^f \geqslant \varepsilon$. The following lemma states an interesting property that holds for mutually dual expressions (cf. [6, Lemma 2]):

Lemma 1. *Let f, g be a pair of mutually dual forms with $|F||G| \geqslant 1$. Then, there exists a variable that occurs either in f or g with frequency at least $1/\log n$, where $n = |F| + |G|$ is the size of f and g.*

Let $x_i \in \{x_1, \ldots, x_N\}$. Then, the expressions f and g can be written as

$$f = x_i f_0(y) \vee f_1(y) \quad \text{and} \quad g = x_i g_0(y) \vee g_1(y),$$

where $y = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N\}$ and f_0, f_1, g_0, and g_1 are the monotone irredundant DNFs with implicant sets

$$F_0 = \bigl\{I \setminus \{i\} \mid i \in I,\ I \in F\bigr\},$$
$$F_1 = \{I \mid i \notin I,\ I \in F\},$$

$$G_0 = \bigl\{J \setminus \{i\} \mid i \in J,\ J \in G\bigr\},$$
$$G_1 = \{J \mid i \notin J,\ J \in G\},$$

respectively. It was shown in [6] that f, g are mutually dual if and only if

$$f_1 \text{ is dual to } g_0 \vee g_1 \quad \text{and} \quad g_1 \text{ is dual to } f_0 \vee f_1. \tag{5}$$

Hence, the initial problem (f, g) of size n is reduced to subproblems

$$(f_1, g_0 \vee g_1) \quad \text{and} \tag{6}$$

$$(g_1, f_0 \vee f_1) \tag{7}$$

of smaller sizes, where x_i acts like a *splitting* variable. If both pairs are mutually dual, then so is the initial one; otherwise a succinct disqualifier can be found.

Fredman and Khachiyan presented an algorithm [6, Algorithm A] that utilizes Lemma 1 and recursively applies decomposition rule (5) to solve MBD in time $n^{O(\log^2 n)}$ for any pair of monotone disjunctive normal forms f and g of size at most n [6, Lemma 4]. As they next show, this running time can be further improved if one notices that subproblems $(f_1, g_0 \vee g_1)$ and $(g_1, f_0 \vee f_1)$ are not independent. Assuming, for example, that subproblem (6) is already solved (and, f_1 is dual to $g_0 \vee g_1$), then the solvability of subproblem (7) is equivalent to the solvability of a system of $|G_0|$ equations

$$g_1\bigl(y[J]\bigr) = f_0\bigl(\bar{y}[J]\bigr), \tag{8}$$

where G_0 is the set of prime implicants of g_0, $J \in G_0$, and $y[J]$ is the vector obtained by y by the substitution $y_j = 1$ for all $j \in J$. However, each of the $|G_0|$ Eq. (8) is equivalent to MBD for the pair of forms (g_1^J, f_0^J) where g_1^J is obtained from $g_1(y)$ by setting $y_j = 1, j \in J$, and f_0^J is obtained from $f_0(y)$ by setting $y_j = 0, j \in J$. Thus, the initial problem (f, g) has been decomposed into $|G_0| + 1$ subproblems in total.

A symmetric decomposition holds if one assumes that subproblem (7) is already solved. The initial problem (f, g) can now be decomposed into $|F_0| + 1$ subproblems in total. We next give the decomposition rules, as presented in [6]:

(i) Let f, g be a pair of monotone DNFs of *volume* $v = |F||G|$ and let a variable x_i occur in f with frequency ε_i^f. Then, in polynomial time the MBD problem for $f = x_i f_0(y) \vee f_1(y)$ and $g = x_i g_0(y) \vee g_1(y)$ can be decomposed into subproblem (6) of volume $|F_1||G| \leqslant (1 - \varepsilon_i^f)|F||G| = (1 - \varepsilon_i^f)v$, plus $|G_0|$ subproblems (g_1^J, f_0^J) of volume at most $|F_0||G_1| = \varepsilon_i^f |F||G| \leqslant \varepsilon_i^f v$ each.

The symmetric decomposition rule for g is as follows:

(ii) If a variable x_i occurs in g with frequency ε_i^g, then in polynomial time the MBD problem for (f, g) can be decomposed into subproblem (7) of volume at most $(1 - \varepsilon_i^g)v$, plus $|F_0|$ subproblems (f_1^I, g_0^I) of volume at most $\varepsilon_i^g v$ each.

Finally, property (5) implies that

(iii) The MBD problem for (f, g) can be decomposed into subproblems (6) and (7) of volumes $(1 - \varepsilon_i^f)v$ and $(1 - \varepsilon_i^g)v$, respectively.

The volume $v = |F||G|$ of f and g is an appropriate measure for the size of the input adopted in [6] and facilitates the analysis of the algorithms. We must note here that any variable with positive frequency may be used in the above rules. This is a point where our version of the algorithm differs from the the algorithm of Fredman and Khachiyan. Algorithm B in [6] solves the MBD problem by recursively incorporating the above decomposition rules. After a rather complicated analysis, Fredman and Khachiyan show that Algorithm B solves the MBD problem in quasi-polynomial time $n^{o(\log n)}$:

Theorem 2 [6, Theorem 1]. *The* MONOTONE BOOLEAN DUALITY *problem can be solved in* $n^{4\chi(n)+O(1)}$ *time, where* $\chi(n)^{\chi(n)} = n$.

3. The nondeterministic algorithm

In this section we present a nondeterministic algorithm for checking the dualization of a pair of monotone expressions in disjunctive normal forms.

The deterministic algorithm of [6] applies appropriately the above decomposition rules, until every subproblem that is produced has constant size in which case it can be solved in constant time. In contrast, our nondeterministic version proceeds in *phases*. Each phase starts with a single subproblem on which and on all of its descendant subproblems we apply the same decomposition rules (in a manner explained below) until all produced subproblems have size at most half the size of the initial problem. At this point the duality of the initial problem is equivalent to the duality of all produced subproblems. We next *nondeterministically* select one of the produced subproblems and check its duality in a new phase. The above are repeated until the selected subproblem is reduced to constant size.

We next present the way the decomposition rules are used in each phase:

The Deterministic Phase

Input: a pair of monotone DNFs f and g satisfying the necessary duality condition (1).

Output: a family P of subproblems (i.e., pairs of DNFs) each of size at most half the size of the input such that the input pair is dual if and only if all pairs in the family are dual.

(1) Delete all redundant implicants from F and G and set $n = |F| + |G|$ and $v = |F||G|$.

(2) Check conditions (2), (3), and (4). If any of these conditions is violated, then f, g are not mutually dual and a succinct disqualifier can be found in polynomial time.

(3) If $\min\{|F|, |G|\} \leqslant 2$, the MBD problem can be solved in polynomial time.

(4) Select a variable x_i such that

$$\varepsilon_i^f \geqslant 1/\log n \quad \text{or} \quad \varepsilon_i^g \geqslant 1/\log n.$$

Comment: Such a variable can be found in polynomial time and its existence follows from Lemma 1.

(5) If $\varepsilon_i^f \leqslant \frac{1}{2}$, apply decomposition rule (i) to obtain one subproblem of volume $(1 - \varepsilon_i^f)v$ plus at most v subproblems of volume at most $\varepsilon_i^f v$ each. Add the produced subproblems to P.

If $\varepsilon_i^f > \frac{1}{2} \geqslant \varepsilon_i^g$, apply decomposition rule (ii) to obtain one subproblem of volume $(1 - \varepsilon_i^g)v$ plus at most v subproblems of volume at most $\varepsilon_i^g v$ each. Add the produced subproblems to P.

If $\varepsilon_i^f > \frac{1}{2}$ and $\varepsilon_i^g > \frac{1}{2}$, apply decomposition rule (iii) to obtain subproblems (6) and (7) of volumes $(1 - \varepsilon_i^f)v$ and $(1 - \varepsilon_i^g)v$, respectively. Add the produced subproblems to P.

(6) Apply steps 1–5 to every subproblem of volume greater than $\frac{1}{2}v$. If such problem does not exist (i.e., the phase has ended), return P.

We next proceed in upper bounding the number of subproblems produced by a phase as a function of the size of the initial problem.

Lemma 3. *The number of the subproblems produced by a* Deterministic Phase *is* $O(n \log n)$, *where* n *is the size of the initial problem.*

Proof. Let (f, g) be the initial pair of expressions of volume v. Firstly assume that variable x_i occurs in f, g with frequencies ε_i^f and ε_i^g, respectively, that are both greater than $\frac{1}{2}$. Then, decomposition rule (iii) is applied and the initial problem is decomposed into two "small" subproblems, one of volume $(1 - \varepsilon_i^f)v \leqslant \frac{1}{2}v$ and another one of volume $(1 - \varepsilon_i^g)v \leqslant \frac{1}{2}v$. Hence, decomposition rule (iii) indicates that the current subproblem will produce no subproblem of volume greater that $\frac{1}{2}v$.

Suppose now that the variable x_i occurs in f with frequency

$$\frac{1}{\log n} \leqslant \varepsilon_i^f \leqslant \frac{1}{2}.$$

According to decomposition rule (i), the initial problem is decomposed into one subproblem of volume $v' = (1 - \varepsilon_i^f)v$ and at most v subproblems of volume at most $\varepsilon_i^f v$ each. Since $\varepsilon_i^f \geqslant 1/\log n$, $v' \leqslant (1 - 1/\log n)v$. Moreover, since $\varepsilon_i^f \leqslant \frac{1}{2}$, the volume of the remaining subproblems is at most $\frac{1}{2}v$. Hence, after decomposition rule (i) is applied, only one "large" (i.e., of volume more than $\frac{1}{2}v$) subproblem is produced that needs further decomposition, while the rest of the subproblems are "small" ones and are not decomposed further until the end of the phase. It is easy to see that the same holds for decomposition rule (ii).

We observe therefore that in all cases at most one subproblem of volume greater than $\frac{1}{2}v$ may result from a problem of size v. If such a problem is produced, we next apply decomposition rule (i) or (ii) to this unique subproblem which has size $n' < n$ and volume $v' \leqslant (1 - 1/\log n)v$ for some variable x_j of frequency, say, ε_j. After decomposition rule is applied, at most $1 + v$ subproblems are produced, one of volume $v'' = (1 - \varepsilon_j)v'$ and at most $v' \leqslant v$ of volume at most $\varepsilon_j v'$ each. See that $\varepsilon_j v' < \frac{1}{2}v$, while

$$\begin{aligned} v'' = (1-\varepsilon_j)v' &\leqslant \left(1 - \frac{1}{\log n'}\right)\left(1 - \frac{1}{\log n}\right)v \\ &\leqslant \left(1 - \frac{1}{\log n}\right)^2 v. \end{aligned}$$

The current phase lasts until the volume of every subproblem produced is at most $\frac{1}{2}v$. The number of decomposition steps required for the phase to complete is $\mathrm{O}(k)$ where k is the solution of the equation:

$$\left(1 - \frac{1}{\log n}\right)^k v = \frac{1}{2}v \quad \Rightarrow$$

$$k = \frac{-\ln 2}{\ln(1 - \frac{1}{\log n})} = \frac{-\ln 2 \log n}{\ln(1 - \frac{1}{\log n})^{\log n}}.$$

When n tends to infinity, $k = \mathrm{O}(\log n)$. Thus, the number of decomposition steps required for the phase to complete is $\mathrm{O}(\log n)$ while $\mathrm{O}(n)$ subproblems are produced at each step. Consequently, at the end of the phase there will be $\mathrm{O}(n \log n)$ subproblems, each of them of volume at most $\frac{1}{2}v$. □

The idea is now to call a nondeterministic oracle to guess the next subproblem for the algorithm to proceed (and the next phase to start). This nondeterministic guessing is used only at the end of each phase. Since at the end of the phase there are $\mathrm{O}(n \log n)$ subproblems in total, the number of nondeterministic guesses required is $\mathrm{O}(\log(n \log n)) = \mathrm{O}(\log n)$.

Observe now that the number of phases is $\mathrm{O}(\log v) = \mathrm{O}(\log n)$ since each phase and the subsequent nondeterministic guessing result in a problem at most half the size of the initial one. Thus, the total number of nondeterministic guesses is $\mathrm{O}(\log^2 n)$. We next summarize the whole algorithm:

The Nondeterministic Algorithm

Input: a pair of monotone DNFs f and g satisfying the necessary duality condition (1).

(1) Nondeterministically guess $\mathrm{O}(\log^2 n)$ bits and store them.
(2) Apply the Deterministic Phase for the current problem (f_c, g_c) and let P be the family of the produced subproblems.
Comment: At the first run of the algorithm, $(f_c, g_c) = (f, g)$.
(3) Utilize the first $\log|P|$ bits stored in step 1 to identify a subproblem in P and make it the next current problem (f_c, g_c). Delete these bits from the stored sequence of bits.
(4) Go to step 2.

We have thus proved the following theorem:

Theorem 4. *The* Nondeterministic Algorithm *solves* MONOTONE BOOLEAN DUALITY *in polynomial time plus* $\mathrm{O}(\log^2 n)$ *nondeterministic guesses, where n is the size of the input.*

Proof. Follows from the above discussion. □

Thus, $\mathrm{O}(\log^2 n)$ nondeterministic guesses suffices to find a succinct disqualifier and prove that the input pair of monotone DNF expressions are not mutually dual. This result places MBD to the subclass of co-NP, the class co-NP$[\log^2 n]$ where only the first $\log^2 n$ are nondeterministic.

Theorem 5. MONOTONE BOOLEAN DUALITY *is in* co-NP$[\log^2 n]$.

Proof. Follows from the definition of co-NP$[\log^2 n]$ and Theorem 4. □

The same result was independently given by Eiter et al. in [4]. Their work is also based on the algorithms of Fredman and Khachiyan, as our. The main difference, however, is that Eiter et al. use the algorithms presented in [6] without any modifications while our work uses the supporting theory and mainly the decomposition rules (in the way presented above) in order to obtain a simpler proof for the $\mathrm{O}(\log^2 n)$ nondeterministic bound. Eiter et al. work on the recursion tree T generated by Algorithm A of [6], they observe that every node (subproblem) α in T is uniquely determined by a path (in other words, by a sequence of left and right moves) from the root (initial problem) to α. They prove that this sequence is obtainable in polynomial time plus $O(\log^3 n)$ suitably guessed bits and thus place MBD in co-NP[$\log^3 n$]. The more restricted bound of $O(\log^2 n)$ nondeterministic guesses was given by Eiter et al. after a more involved proof applied on Algorithm B of [6].

4. Conclusions

In this work we presented a nondeterministic algorithm for checking the duality of a pair of monotone expressions in disjunctive normal form. The algorithm utilizes the decomposition rules given by Fredman and Khachiyan in [6] as well as the appearance of variables with appropriate frequency. Our algorithm requires deterministic polynomial time plus $O(\log^2 n)$ nondeterministic steps. This result places the monotone Boolean dualization in co-NP[$\log^2 n$], the subclass of co-NP where only the first $\log^2 n$ steps are nondeterministic. It also makes straightforward the quasi-polynomial running time of the deterministic version, avoiding the rather complicated analysis presented in [6]. Future work includes further investigation on the exact time complexity of the monotone Boolean dualization, a long unresolved question.

Acknowledgements

Research by ECS was supported by the University of Patras Research Committee (Project Caratheodory under contract no. 1939). The authors wish to thank the corresponding editor for pointing out reference [4].

References

[1] R. Beigel, B. Fu, Molecular computing, bouned nondeterminism, and efficient recursions, Algorithmica 25 (1999) 222–238.

[2] J.C. Bioch, T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, Inform. and Comput. 123 (1995) 50–63.

[3] J. Diaz, J. Torán, Classes of bounded nondeterminism, Math. Systems Theory 23 (1990) 21–32.

[4] T. Eiter, G. Gottlob, K. Makino, New results on monotone dualization and generating hypergraph transversals, in: Proc. of 34th ACM Symposium on Theory of Computing, Montreal, Quebec, Canada, May 19–21, 2002.

[5] T. Eiter, G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, SIAM J. Comput. 24 (6) (1995) 1278–1304.

[6] M.L. Fredman, L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms, J. Algorithms 21 (1996) 618–628.

[7] J. Goldsmith, M.A. Levy, M. Mundhenk, Limited nondeterminism, ACM SIGACT News 27 (2) (1996) 20–29.

[8] D. Gunopulos, R. Khardon, H. Mannila, H. Toivonen, Data mining, hypergraph transversals, and machine learning, in: Proc. of 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Tucson, AZ, May 12–14, 1997, pp. 209–216.

[9] V. Gurvich, L. Khachiyan, On generating the irredundent conjunctive and disjunctive normal forms of monotone Boolean functions, Technical Report LCSR-TR-251, Department of Computer Science, Rutgers University, 1995.

[10] D.S. Johnson, M. Yannakakis, C.H. Papadimitriou, On generating all maximal independent sets, Inform. Process. Lett. 27 (1988) 119–123.

[11] D.J. Kavvadias, M. Sideri, E.C. Stavropoulos, Generating all maximal models of a Boolean expression, Inform. Process. Lett. 74 (3–4) (2000) 157–162.

[12] D.J. Kavvadias, E.C. Stavropoulos, Evaluation of an algorithm for the transversal hypergraph problem, in: J.S. Vitter, C.D. Zaroliagis (Eds.), Proc. of 3th International Workshop on Algorithm Engineering (WAE99), London, UK, in: Lecture Notes in Comput. Sci., Springer, Berlin, 1999, pp. 72–84.

[13] C.M.R. Kintala, P. Fischer, Refining nondeterminism in relativized polynomial-time bounded computations, SIAM J. Comput. 9 (1980) 46–53.

[14] C.H. Papadimitriou, NP-completeness: A retrospective, in: Proc. of 24th International Colloquium on Automata, Languages, and Programming, Bologna, Italy, July 7–11, 1997, pp. 2–6.