Electronic Colloquium on Computational Complexity, Report No. 139 (2013)

Detecting Monomials with k Distinct Variables

Peter Floderus[1], Andrzej Lingas *[2], Mia Persson[3], and Dzmitry Sledneu[1]

[1] The Centre for Mathematical Sciences, Lund University, 22100 Lund, Sweden.
Peter.Floderus@maths.lth.se, Dzmitry@maths.lth.se
[2] Department of Computer Science, Lund University, 22100 Lund, Sweden. Andrzej.Lingas@cs.lth.se
[3] Department of Computer Science, Malmö University, 20506 Malmö, Sweden. Mia.Persson@mah.se

Abstract. We study the complexity of detecting monomials with special properties in the sum-product expansion of a polynomial represented by an arithmetic circuit of size polynomial in the number of input variables and using only multiplication and addition. We focus on monomial properties expressed in terms of the number of distinct variables occurring in a monomial. Our main result is a randomized FPT algorithm for detection of a monomial having at least k distinct variables, parametrized by the degree of the polynomial. Furthermore, we derive several hardness results on detection of monomials with such properties within exact, parametrized and approximation complexity. In particular, we observe that the detection of a monomial having at most k distinct variables is $W[2]$-hard for the parameter k.

Keywords: polynomial, arithmetic circuit, parametrized complexity, approximation hardness

1 Introduction

Koutis initiated a group algebra approach to derivation of fully parametrized algorithms for combinatorial problems in [6]. He considered the k-path problem, which is to determine if a graph contains a simple path of length at least k, and packing problems in [6]. Williams continued this approach in [11] and improved Koutis' parametrized upper time bound for the k-path problem to $O^*(2^k)$ (the O^* notation suppresses polynomial in n and k factors). The aforementioned bounds have been obtained by a fixed-parameter reduction to the detection of the so called multilinear monomial in the sum-product expansion of a polynomial represented by arithmetic circuit of polynomial size using only addition and multiplication. A monomial is *multilinear* if no variable occurs in it more than once. Williams showed that a multilinear monomial in the sum-product expansion of the aforementioned polynomial can be detected in $O^*(2^l)$ time, where l is the degree of the polynomial (Theorem 3.1 in [11]). In the subsequent paper [8], Koutis and Williams derived new fixed-parameter upper time bounds for several other combinatorial problems by fixed-parameter reductions to the problem of multilinear monomial detections. Chen et al. generalized the FPT method of Koutis and Williams to detecting a monomial in which no variable occurs p or more times, where p is a prime number, parametrized by the degree of the polynomial [2]. An extension of the generalization to include non-necessarily prime p was given in [3].

In this paper, we continue the topic of detecting monomials having some special property in the sum-product expansion of a polynomial in n variables, represented by an arithmetic circuit of size polynomial in n, using only addition and multiplication. We focus on special properties of a monomial expressed in terms of the number of distinct variables occurring in it.

We begin with a technical lemma on decomposition of a polynomial and a monotone circuit representing it into uniform polynomials (i.e., polynomials whose all monomials have the same degree) and representing them monotone circuits.

* Research supported in part by Swedish Research Council grant 621-2011-6179.

ISSN 1433-8092

Our main result is a randomized FPT algorithm for detection of a monomial having at least k distinct variables, parametrized by the degree of the polynomial. It relies on a slight generalization of the group algebra method from [6, 11] given in [8]. Our algorithm runs in $O^*(2^k)$ time, where the O^* notation suppresses polynomial in the number of input variables and the degree of the polynomial factors. It yields also an FPT algorithm for the decision version of the max q-cover problem [5] with subsets of size not exceeding s, with respect to the parameters q and s.

We can also provide a deterministic FPT algorithm for detection of a monomial having at most k distinct variables, parametrized by the degree of the polynomial, provided that in the circuit representing the polynomial no multiplication gate is a descendant of a multiplication gate.

Next, we observe that detecting a monomial with at most k distinct variables is $W[2]$-hard with respect to the parameter k while detecting a monomial with at least n distinct variables is $W[2]$-hard with respect to the degree of the polynomial divided by n.

We also consider the problem of finding the smallest k such that the polynomial has a monomial with at most k distinct variables as well as the problem of finding the largest k such that the polynomial has a monomial with at least k distinct variables. We show that the minimization problem cannot be approximated within $O(2^{\log^{1-\epsilon} n})$ for any $\epsilon > 0$, unless $NP \subset DTIME(n^{poly\log(n)})$ while the maximization problem cannot be approximated within $1 - \frac{1}{e} + \epsilon$ for any $\epsilon > 0$, unless $NP = P$.

As a by-product, we also obtain hardness results on detection of implicants of Boolean functions represented by monotone Boolean circuits, with at most (or, at least, respectively) k distinct variables. We infer in particular that deciding whether or not two monotone Boolean circuits compute different functions is NP-complete.

Organization. In the next section, we define a monotone arithmetic circuit, a monotone Boolean circuit, and state a lemma from [8] generalizing the main results from [6, 11]. In Section 3, we present and analyze our FPT algorithm while in Section 4, we show our hardness results.

2 Preliminaries

A monotone arithmetic circuit is a directed acyclic graph where each leaf is labeled either with a variable or a real constant (input gates), each non-leaf vertex has fan-out two and it is labeled with either $+$ (addition gate) or $\times$ (multiplication gate), and a single vertex is distinguished as an output gate. A monotone Boolean circuit is defined analogously, with $\vee$ labels instead of $+$ labels, $\wedge$ labels instead of $\times$ labels and Boolean $0, 1$ constants instead of real constants.

For a positive integer k, the set of positive integers not greater than k will be denoted by $[k]$.

In Lemma 1 in [8], Koutis and Williams provided a slight generalization of their main results from [6, 11]. We can rephrase the generalization in terms of our notation as follows.

Fact 1 ([8]). *Let $P(x_1, ..., x_n, z)$ be a polynomial represented by a monotone arithmetic circuit of size $s(n)$. There is a randomized algorithm that for every P runs in $O^*(2^k t^2 s(n))$ time and outputs "YES" with high probability if there is a monomial of the form $z^t Q(x_1, ..., x_n)$, where $Q(x_1, ..., x_n)$ is a multilinear monomial of degree at most k, in the sum-product expansion of P, and always outputs "NO" if there is no such monomial $z^t Q(x_1, ..., x_n)$ in the expansion.*

3 FPT algorithms

3.1 Detecting a monomial with at least k variables

In this section, we present an FPT algorithm with respect to the polynomial degree for the "at least k distinct variables" monomial property.

Our FPT algorithm relies on Fact 1 and the following lemma on decomposition of a polynomial and a monotone circuit into uniform polynomials (i.e., polynomials whose all monomials have the same degree) and representing them circuits. The lemma should be of interest in its own rights.

Lemma 1. *Let C be a monotone arithmetic circuit of size $s(n)$ representing a polynomial of maximum degree l. One can construct a monotone arithmetic circuit C^* of size $O(l^2 s(n))$ such that for each gate g in C there is a sequence of gates $g_0, ..., g_l$ in C^*, where for $j = 0, ..., l$, the sum-product expansion of the polynomial represented by g_j consists of all monomials of degree j in the sum-product expansion of the polynomial represented by g. The construction takes $O(l^2 s(n))$ time.*

Proof. We construct the sequences of gates $g_0, ..., g_l$ in C^* corresponding to gates g in C by induction on the height of g in C, i.e., the maximum distance of g to an input gate in C. If g is a leaf gate the construction is trivial. Suppose that g is an addition or multiplication gate getting input from gates f and h in C. Then, if g is an addition gate, for $j = 0, ..., l$, we set g_j to an addition gate getting inputs from f_j and h_j. Otherwise, if g is a multiplication gate then for $j = 0, ..., l$, we insert for $s = 0, ..., j$, intermediate multiplication gates $g_{j,s}$ computing the products of f_s with h_{j-s}, and also $O(j)$ intermediate addition gates so at g_j the sum of outputs of these intermediate multiplication gates is computed. Thus, for the construction of each sequence of gates $g_0, ..., g_l$ in C^*, we use $O(l^2)$ gates. □

Theorem 1. *Let l be a natural number and let k be a natural number not larger than l. Next, let $P(x_1, ..., x_n)$ be a polynomial of maximum degree l represented by a monotone arithmetic circuit of size $s(n)$. There is a randomized algorithm that for every P runs in $O^*(2^k s(n))$ time and outputs "YES" with very high probability if there is a monomial with at least k distinct variables in the sum-product expansion of P, and always outputs "NO" if there is no such monomial in the expansion.*

Proof. To begin with, we modify the circuit C to the circuit D by creating a new input gate labeled with z, and n addition gates representing $x_i + z$, $i = 1, ..., n$, respectively, and for $i = 1, ..., n$, connecting each direct ancestor of the input gate labeled with x_i with the addition gate representing $x_i + z$, respectively. Thus, the circuit D represents the polynomial $P(x_1 + z, ..., x_n + z)$.

Note that each monomial in the sum-product expansion of $P(x_1 + z, ..., x_n + z)$ can be obtained from some monomial in the sum-product expansion of $P(x_1, ..., x_n)$ by substitution of z for some occurrences of some variables x_i. Hence, there is a monomial with at least k distinct variables in the sum-product expansion of $P(x_1, ..., x_n)$ if and only if there is a monomial of the form $z^t Q(x_1, .., x_n)$, where $Q(x_1, .., x_n)$ is a multilinear monomial with precisely k distinct variables in $\{x_1, ..., x_k\}$, in the sum-product expansion of $P(x_1 + z, ..., x_n + z)$.

In the next step, we construct the monotone arithmetic circuit D^* on the basis of the input circuit D by using Lemma 1. D^* has $O(l^2 s(n))$ gates and its construction takes time linear in its size. For each $r = 1, ..., l$, D^* has a gate o_r such that the polynomial $O_r(x_1, ..., x_n, z)$ generated at o_r can be represented as the sum of all monomials of degree r in the sum-product expansion of the polynomial $P(x_1 + z, ..., x_n + z)$.

Finally, we iterate the following step for $r = k, ..., l$. We run the FPT algorithm due to Koutis and Williams [8] given in Fact 1 on the sub-circuit of D^* representing the polynomial $O_r(x_1, ..., x_n, z)$ in order to determine if the sum-product expansion of $O_r(x_1, ..., x_n, z)$ contains a monomial of the form $z^{r-k} Q(x_1, .., x_n)$, where $Q(x_1, .., x_n)$ is multilinear monomial with precisely k distinct variables in $\{x_1, ..., x_k\}$. Whenever, we obtain a positive answer, we report "YES" and stop.

By Fact 1, such a run of the FPT algorithm due to Koutis and Williams [8] takes $O^*(2^k l^2 size(D^*))$ time, which is $O^*(2^k l^4 s(n))$ by Lemma 1. In order to amplify the probability of the "YES" answer

if there is a monomial with at least k distinct variables in the sum-product expansion of P, we can replace each run of the algorithm of Williams and Koutis from [8], with $O(\log n)$ runs. □

Applications. The max q-cover problem [5] is as follows: for given subsets $S_1, S_2, ..., S_m$ of a ground set X, find a subfamily of $\{S_1, ..., S_m\}$ containing at most q subsets maximizing the number of elements in X covered by the subsets included. In the decision version of this problem, there is also given a positive integer k and the objective is to decide if there is a subfamily of $\{S_1, ..., S_m\}$ containing at most q subsets whose union covers at least k elements.

Theorem 2. *The decision version of the max q-cover problem with all given subsets of size bounded by s admits an FPT algorithm with respect to the parameters q and s.*

Proof. For $i = 1, ..., |X|$, associate the variable x_i with the i-th element in X. We shall denote elements in X by their numbers. Next, for $j = 1, ..., m$, let M_j denote the monomial $\prod_{i \in S_j} x_i$. It is clear that a monomial of the polynomial $\left(\sum_{j=1}^{m} M_j\right)^q$ with at least k distinct variables corresponds to a union of at most q subsets S_j covering at least k elements in X and *vice versa*. Since the degree of the polynomial does not exceed qs, the theorem follows from Theorem 1.

□

Note that the known FPT algorithms for the hitting set with bounded set size [4, 9] translate to FPT algorithms for set cover with a bound on maximum number of sets that can cover a single element.

3.2 Detecting a monomial with at most k variables

In this subsection, we present an FPT algorithm with respect to the polynomial degree for the "at most k distinct variables" monomial property for polynomials represented by restricted monotone arithmetic circuits. The restriction does not allow any multiplication gate to be a descendant of an addition gate. We shall call a monotone arithmetic circuit obeying this requirement an *addition-multiplication circuit*.

Theorem 3. *Let l be a natural number and let k be a natural number not larger than l. Next, let $P(x_1, ..., x_n)$ be a polynomial of maximum degree l represented by an addition-multiplication circuit of size $s(n)$. There is an algorithm that for every P runs in $O^*(2^{2^d})$ time and outputs "YES" if there is a monomial with at most k distinct variables in the sum-product expansion of P, and otherwise outputs "NO".*

Proof. Note that $P(x_1, ..., x_n)$ has a decomposition of the form $\prod_{i=1}^{l} \sum_{j=1}^{l_i} x_{a(i)_j}$. Importantly, for each of the sums, we can determine a distinct addition gate in the circuit representing it, and consequently the variables in the sum.

We shall reduce the problem of detecting a monomial of P having at most k distinct variables to the problem of minimum set cover over a universe $U = \{u_1, ..., u_l\}$, with l elements as follows. We associate the i-th element u_i of the universe with the i-th sum. On the other hand, with the variable x_m, we associate the subset S_m of U, which contains u_i iff the variable x_m occurs in the i-th sum.

It is easy to see that for an arbitrary monomial of P, the subsets S_m associated with the variables x_m occurring in the monomial form a set cover of U, and conversely, for any set cover of U composed of the subsets S_m, there is a monomial of P composed exactly of the corresponding variables (some of them can occur many times in the monomial).

Note also that there are at most 2^l different subsets S_m. Therefore, we can solve the minimum set cover problem for the aforementioned instance in $O^*(2^{2^l})$ time by brute force. The set of variables x_m corresponding to the subsets S_m in the minimum set cover yields a minimum cardinality set of variables that occur in a monomial of P. □

4 Hardness results

Since the decision version of the set cover problem can be easily encoded as a problem of detecting a monomial with at most k variables, we obtain the following theorem.

Theorem 4. *Let P be a polynomial in n variables represented by a monotone arithmetic circuit with $O(n^2)$ gates. The problem of deciding if P has a monomial with at most k distinct variables is NP-complete as well as $W[2]$-hard for the parameter k.*

Proof. Consider an instance of the set cover problem with ground set X, a family of subsets $S_1, ..., S_n$ of X whose union covers X, and a positive integer k. We may assume w.l.o.g. that $|X| \leq n$ since the $W[2]$-hardness of set cover follows from that of dominating set [4]. For $j = 1, ..., n$, associate with S_j the variable y_j. Let PS stand for the polynomial $\prod_{x \in X}(\sum_{j \ s.t. \ x \in S_j} y_j)$. Note that the polynomial PS has n variables and it can be represented by a monotone arithmetic circuit of size $O(n^2)$. It is clear that X can be covered with $\leq k$ of the subsets $S_1, ..., S_n$ iff PS has a monomial with $\leq k$ distinct variables. This many-one reduction is clearly fixed-parameter with respect to k. Since the set cover problem is NP-complete and $W[2]$-complete with respect to k, we conclude that the decision version of the minimum-variable monomial problem is NP-complete and $W[2]$-hard for the parameter k.

□

Chechik et al. [1] have recently studied among other things the following *secluded path problem*: for a graph and its two vertices, find a path connecting the two vertices that minimize the number of neighbors of vertices on the path. Note that in particular all vertices on such a path are accounted to the set of the path neighbors. They proved among other things that the secluded path problem is NP-hard and, unless $NP \subset DTIME(n^{poly \log(n)})$, inapproximable within a factor of $O(2^{\log^{1-\epsilon} n})$ for any $\epsilon > 0$.

For a vertex v in a graph $G = (V, E)$, let $N(v)$ denote the (closed) neighborhood of v, i.e., the set of all vertices adjacent to v in G, augmented by v. A *neighborhood walk* in G is a sequence of vertex neighborhoods $N(v_1), N(v_2), ..., N(v_l)$ in G such that for $j = 1, ..., l-1$, $\{v_j, v_{j+1}\} \in E$. The length of the walk is $l - 1$.

We define recursively the polynomial $Q_l(i, j)$ whose monomials are in one-to-one correspondence with neighborhood walks of length $l - 1$ in G starting from $N(v_i)$ and ending with $N(v_j)$ as follows:

$$Q_1(i, i) = \prod_{v_k \in N(v_i)} x_k$$

$$Q_l(i, j) = \sum_{v_q \in N(v_j)} Q_{l-1}(i, q) \prod_{v_k \in N(v_j)} x_k$$

Two following lemmata are straightforward.

Lemma 2. *There is a neighborhood walk of length $l-1$ starting from $N(v_i)$ and ending with $N(v_j)$ in G such that the union of its vertex neighborhoods has cardinality at most k iff $Q_l(i,j)$ has a monomial having at most k different variables.*

Lemma 3. *For a graph on n vertices, $i,j,l \in [n]$, the polynomial $\sum_{l=1}^{n-1} Q_l(i,j)$ can be represented by a monotone arithmetic circuit of size $O(n^2)$. Furthermore, there is a path in G with at most k neighbors starting from v_i and ending with v_j iff the polynomial $\sum_{l=1}^{n-1} Q_l(i,j)$ has a monomial having at most k different variables.*

Lemma 3 yields the following theorem.

Theorem 5. *Let P be a polynomial in n variables represented by a monotone arithmetic circuit with $O(n^c)$ gates. For $c \geq 2$, the problem of determining the smallest k such that P has a monomial with at most k distinct variables cannot be approximated within $O(2^{\log^{1-\epsilon} n})$ for any (fixed) $\epsilon > 0$, unless $NP \subset DTIME(n^{poly\log(n)})$. On the other hand, for any positive constant c, it can be solved exactly in $O(n^{k_0+c})$ time, where k_0 is the minimum number of distinct variables in a monomial of P.*

Proof. Lemma 3 yields a many-one fixed-parameter reduction from the secluded path problem to the problem of determining the smallest k such that a polynomial in n variables represented by a circuit of polynomial size has a monomial with precisely k distinct variables. Let m be a minimum number of neighbors on a path connecting two vertices v_i and v_j of a graph G on n vertices. By Lemma 3, if the aforementioned minimum-variable monomial problem had an $O(2^{\log^{1-\epsilon} n})$ approximation algorithm then we could answer that there is a path connecting this pair of vertices and having at most $O(m/2^{\log^{1-\epsilon} n})$ neighbors (importantly, observe that that the number n of variables in the polynomial $\sum_{l=1}^{n-1} Q_l(i,j)$ is equal to the number of vertices in G). This would contradict the lower bound on the approximability of the secluded path problem established in [1].

We can obtain an exact solution to the minimum -variable monomial problem by simply evaluating the polynomial $\sum_{l=1}^{n-1} Q_l(i,j)$ for $q = 1, 2, ...$ on all assignments of zero-one values with q ones until a non-zero value is produced. □

By substituting $\vee$ gates for $+$ gates and $\wedge$ gates for $\times$ gates in an arithmetic circuit, we obtain a corresponding monotone Boolean circuit (i.e., a directed acyclic graph whose non-leaf nodes are labeled with either $\vee$ or $\wedge$, leaves are labeled with distinct variables, and a single node is distinguished as an output node). It computes a Boolean function which is a disjunction of Boolean monomials, called *implicants*, corresponding to the monomials of the polynomial represented by the original arithmetic circuit. Hence, we can derive the following corollary and theorem from Theorem 5.

Corollary 1. *Given a monotone Boolean circuit with n input variables, and $O(n^2)$ gates, the problem of determining the smallest k such that the Boolean function computed by the circuit has an implicant with precisely k distinct variables cannot be approximated within $O(2^{\log^{1-\epsilon} n})$ for any $\epsilon > 0$, unless $NP \subset DTIME(n^{poly\log(n)})$. The decision version of this problem is NP-complete.*

Theorem 6. *Given two monotone Boolean circuits with the same set of n input variables, $O(n^2)$ gates, the problem of determining if the Boolean functions computed by them are different is NP-complete.*

Proof. Let B be the monotone Boolean function computed by a given monotone Boolean circuit with n variables $x_1, ..., x_n$ and a single output node. Next, for $1 \leq l < n$, let D_l be the Boolean function that is a disjunction of all monotone implicants with precisely l distinct variables in $\{x_1, ..., x_n\}$.

D_l can be easily computed by a monotone Boolean circuit of polynomial size as follows. Let D_l^j be the Boolean function that is a disjunction of all monotone implicants with precisely l distinct variables in $\{x_1, ..., x_j\}$. For $1 < l$ and $j < n$, we have $D_l^j = D_{l-1}^{j-1} \wedge x_j \vee D_l^{j-1}$. Now, let $k < n$ and $C = B \vee D_{k+1}$. Note that $C \neq D_{k+1}$ iff B has an implicant composed of at most k distinct variables. The theorem follows from Corollary 1. □

Theorem 6 is interesting since no negation is used by any of the two circuits, otherwise the theorem would follow trivially from the NP-completeness of the satisfiability problem.

By a reduction from the max q-cover problem (see Applications in Section 3.1), we can obtain also similar although weaker results on inapproximability of the symmetric problem when monomials or implicants with maximum number of distinct variables are sought.

Theorem 7. *Let P be a polynomial in n variables represented by a monotone arithmetic circuit with polynomial in n number of gates. The problem of determining the largest k such that P has a monomial with precisely k distinct variables cannot be approximated within $1 - \frac{1}{e} + \epsilon$ for any $\epsilon > 0$, unless $NP = P$. The decision version of this problem is NP-hard.*

Proof. sketch. Use the notation and the polynomial $\left(\sum_{j=1}^{m} M_j\right)^q$ from the proof of Theorem 2. A monomial of this polynomial with the maximum number of distinct variables corresponds to a union of at most q subsets S_j covering the same maximum number of elements in X and *vice versa*. This combined with the fact that max q-cover cannot be approximated within $1 - \frac{1}{e} + \epsilon$ for any $\epsilon > 0$, unless $NP = P$ (see [5]) yields the theorem.

□

Corollary 2. *Given a monotone Boolean circuit with n input variables and polynomial in n size, the problem of determining the largest k such that the Boolean function computed by the circuit has an implicant with precisely k distinct variables cannot be approximated within $1 - \frac{1}{e} + \epsilon$ for any $\epsilon > 0$, unless $NP = P$. The decision version of this problem is NP-hard.*

If we set k to n in the proof of Theorem 7, then we obtain a fixed-parameter reduction of the set cover problem to the problem of detecting if the constructed polynomial of degree $O(nq)$ has a monomial with precisely n distinct variables, with respect to the parameter q. Since the set cover problem is $W[2]$-complete, we obtain the following theorem.

Theorem 8. *Let P be a polynomial with n variables and maximum degree l represented by a monotone arithmetic circuit with polynomial in n number of gates. The problem of determining if P has a monomial with precisely n distinct variables is $W[2]$-hard for the parameter l/n.*

References

1. S. Chechik, M.P. Johnson, M. Parter, and D. Peleg, *Secluded Connectivity Problems*, Proc. of ESA 2013, LNCS, pages 301-313.
2. S. Chen, *Monomial testing and applications*, Proc. of Frontiers in Algorithmics and Algorithmic Aspects in Information and Management 2013. arXiv:1303.0478v2
3. Z. Chen and B. Fu and Y. Liu and R. Schweller, *Algorithms for testing monomials in multivariate polynomials.* Proc. of COCOA 2011. arXiv:1007.2675
4. R.G. Downey and M.R. Fellows, Parametrized Complexity, Springer, 1999, New York.
5. U. Feige, *A Threshold of* $\ln n$ *for Approximating Set Cover*, J. ACM 45, pp. 634-652.
6. I. Koutis, *Faster algebraic algorithms for path and packing problems*, Proc. of ICALP 2008, LNCS 5125, pp. 575-586 (2008).
7. I. Koutis, *A faster parametrized algorithm for set packing,* Information Processing Letters, 94(1) (2005), pp. 4–7.
8. I. Koutis and R. Williams, *Limits and Applications of Group Algebras for Parameterized Problems,* Proc. of ICALP (1) 2009, LNCS, pp. 653-664 (2009).
9. R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006, New York.
10. L.G. Valiant, *Why is boolean complexity difficult?*, Boolean Function Complexity, Lond. Math. Soc. Lecture Note Ser. vol 169.
11. R. Williams, *Finding paths of length k in* $O^*(2^k)$ *time*, Information Processing Letters, 109(6) (2009), pp. 315–318.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de