REPORTS
IN
INFORMATICS

ISSN 0333-3590

On solving sparse algebraic equations over finite fields

Igor Semaev

REPORT NO 308 **September 2005**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL
http://www.ii.uib.no/publikasjoner/texrap/pdf/2005-308.pdf

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

On solving sparse algebraic equations over finite fields

Igor Semaev
igor@ii.uib.no
Department of Informatics, University of Bergen, Norway

13th September 2005

Abstract

A system of algebraic equations over the finite field F_q is called sparse if each equation depends on a small number of variables. In this paper new algorithms for solving such equations are presented. The mathematical expectation of their running time is derived. It compares favorably with the worst case estimations coming from the l-SAT problem analysis for $q = 2$ and the trivial estimation with the brute force algorithm for a general q.

1 Introduction

Let F_q be the finite field of q elements and $X = \{x_1, x_2, \ldots, x_n\}$ be a set of variables from F_q. By X_i, $1 \le i \le N$ we denote subsets of X of size $l_i \le l$. The system of equations

$$\begin{cases} f_1(X_1) = 0, \\ \ldots \\ f_N(X_N) = 0 \end{cases} \tag{1}$$

is considered, where f_i are polynomials over F_q and they only depend on variables X_i. Such equations are called l-sparse. One looks at f_i as mappings from the set of all l_i-tuples over F_q to F_q and vice versa any such mapping may be represented by a polynomial over F_q.

Obviously, the equation $f_i(X_i) = 0$ may be also defined by the set V_i of F_q-vectors in variables X_i, called X_i-vectors, on which f_i is zero, so that V_i is defineable by a plenty of polynomials over F_q. We look for the set of all solutions in F_q to the system of equations (1).

When $q = 2$ the equations (1) are Boolean and the decision problem related to such a system of l-sparse algebraic equations is polynomially-time equivalent to the l-satisfiability problem (l-SAT). l-SAT is a problem to determine, given a conjunctive normal form F with n variables and such that each clause of F contains at most l literals, whether or not there is a satisfying assignment for F. Really, let

$$f(x_1, \ldots, x_l) = 0 \tag{2}$$

be any Boolean equation in l Boolean variables and

$$\begin{gathered} (a_{11}, \ldots, a_{1l}), \\ \ldots, \\ (a_{s1}, \ldots, a_{sl}), \end{gathered}$$

be all binary vectors such that $f(a_{i1}, \ldots, a_{il}) = 1$. The vector $(b_1, \ldots, b_l)$ is a solution to (2) if and only if it is a satisfying assignment for the conjunctive normal form

$$F_f = (x_1^{a_{11}} \vee \ldots \vee x_l^{a_{1l}}) \wedge \ldots \wedge (x_1^{a_{s1}} \vee \ldots \vee x_l^{a_{sl}}),$$

where we denote

$$x^a = \begin{cases} x, & \text{if } a = 0, \\ \bar{x}, & \text{if } a = 1, \end{cases}$$

that is $x^a = 0$ if and only if $x = a$. Given the system of equations (1) one constructs a conjunctive normal form F which is a conjunction of F_{f_i}. One now sees that $(b_1, \ldots, b_n)$ is a solution to (1) if and only if this vector is a satisfying assignment for F. Obviously, any l-SAT problem may be represented by a system of l-sparse Boolean equations.

l-SAT (for $l \geq 3$) is one of the classical NP complete problems and there is a vast reference list on this problem. Recently there has been a large effort to design deterministic and randomized exact algorithms for this problem. These efforts resulted in a number of deep and powerful techniques for solving SAT efficiently. Nice examples of such techniques are the Davis-Putnam algorithm, Shöning Local Search and Random Walks. SAT on n variables can be trivially solved in time $O(2^n)$ by trying all possible assignments, but constructing an $O(c^n)$ algorithm for $c < 2$ is a long standing open problem. However, for small values of l there are much faster algorithms for l-SAT with $c = c_l < 2$, see the survey article [5]. Such bounds are worst case estimations to the problem of solving equations (1) when $q = 2$.

In this paper we suggest a new and simple algorithm to solve equations (1), called Gluing Algorithm. Its analysis is based on the following assumptions: given a sequence of natural numbers $l_1, \ldots, l_N \leq l$, the subsets of variables $X_1, \ldots, X_N$ and polynomials (or mappings) $f_1, \ldots, f_N$ are chosen uniformly and independently of each other. In this setting the running time of the Gluing Algorithm is a random variable. We prove that its mathematical expectation is essentially $O((qe^{\gamma_0} + \epsilon)^n)$ operations, when q and l are fixed and n tends to infinity. Here

$$\gamma_0 = -\frac{\ln q}{l} - (q^{\frac{1}{l}} - 1) \ln(\frac{1 - q^{-1}}{1 - q^{-\frac{1}{l}}}),$$

and ϵ is any positive real number. See Theorem 1 for an exact statement. This bound compares favorably with the trivial bound $O(q^n)$ for a general q and the worst case bounds coming from the l-SAT problem analysis when $q = 2$ as one sees from the data tabulated below:

	the worst case	Gluing1, the average	Gluing2, the average
c_3	1.324	1.262	1.238
c_4	1.474	1.355	1.326
c_5	1.569	1.425	1.393
c_6	1.637	1.479	1.446
...	...	...	...

The constant c_l is given such that the algorithm runs in time $O(c_l^n)$. The Gluing1 Algorithm is a modification of the Gluing Algorithm with the same running time and minor memory requirements. The Gluing2 Algorithm is even faster, see Theorem 2, but the amount of memory used is of the same order of magnitude as its running time.

This article was motivated by applications in cryptanalysis. Modern ciphers are product, that is the mappings they implement are compositions of functions in small number of variables. Then intermediate variables are introduced to simplify equations, describing the cipher, and to get a system of sparse equations. Solving this system of equations breaks the cipher. Previously known approaches to solve such equations are mostly efficient for an

overdefined system of equations of low algebraic degree, see [1],[2] and [4]. Here we are studying an approach which exploits the sparsity of equations and does depend neither on their algebraic degree nor whether the system is overdefined or not. It seems that Håvard Raddum was the first to explore this approach in [8], though his methods differ from ours. The Gluing Algorithm and its modifications presented in this paper have not been able to undermine modern ciphers so far, as the number of intermediate variables is usually too big, but some combinations with other techniques proved to be promising in cryptanalysis, [9]. Getting asymptotic bounds on the complexity of these latter combinations is in progress.

The rest of the paper is organized as follows. In Section 2 we describe the Gluing and Gluing1 Algorithms and in Section 3 asymptotic bounds on their running time are given. In Section 4 we explain the Gluing2 Algorithm and prove a better bound for its running time. Section 5 presents some auxiliary results on the distribution of random variables related to random allocations used in the previous Sections.

The author thanks Fedor Fomin and Håvard Raddum for usefull discussions.

2 Algorithms

We'll describe the Gluing procedure. Given pairs (X_i, V_i) for $i = 1$ and 2 one defines the sets of variables $Z = X_1 \cup X_2$ and $Y = X_1 \cap X_2$ and the set of Z-vectors U by the following rule

$$U = \{(a_1, b, a_2)/(a_1, b) \in V_1, (b, a_2) \in V_2\},$$

where a_i is an $(X_i \setminus Y)$-vector and b is an Y-vector. We see that to glue (X_1, V_1) and (X_2, V_2) one can sort sets V_1 and V_2 by Y-subvectors and only glues vectors with the same Y-subvector. So the complexity of the gluing is on the whole

$$O(|U| + |V_1|log|V_1| + |V_2|log|V_2|) \tag{3}$$

operations like rewriting and comparison with F_q-vectors of size $|Z|$. In the next Section we'll use a simpler bound $O(|V_1||V_2|)$, when estimating the Gluing algorithm, while the bound (3) is actually used in Section 4.

We denote gluing by

$$(Z, U) = (X_1, V_1) \circ (X_2, V_2).$$

and formulate an algorithm to solve the system of equations (1).

Gluing Algorithm

input: the system of equations (1) given by pairs (X_i, V_i), where $1 \leq i \leq N$.

output: the set U of all solutions to (1) in variables $X_1 \cup \ldots \cup X_N$.

put $(Z, U) \leftarrow (X_1, V_1)$ and $k \leftarrow 2$,

while $k \leq N$ do

$(Z, U) \leftarrow (Z, U) \circ (X_k, V_k)$ and $k \leftarrow k + 1$,

return (Z, U).

It is obvious that U is finally the set of all solutions to (1).

The amount of the memory used by the Gluing Algorithm has the same order of magnitude as its running time which is exponential in n, see the next Section for details. This makes the Algorithm impractical. The following variant of the Algorithm requires $O(\text{poly}(n))$ bits of memory.

In the description below we will use some new notation. For an X_1-vector a and an X_2-vector b we denote by $a \circ b$ an $X_1 \cup X_2$-vector which is the result of the gluing procedure applied to a and b. This is only possible when a and b have the same $X_1 \cap X_2$-subvector. We also suppose the each set V_i to be sorted according to some order $\leq$, which may depend on i. Let a be an X-vector and Y be a subset of X then a_Y denotes the Y-subvector of a(this is the projection of a on Y). Finally, $X(k) = X_1 \cup \ldots \cup X_k$ and $X(0) = \emptyset$, $a_\emptyset = \emptyset$, $a \circ \emptyset = a$.

Gluing1 Algorithm

input: the system of equations (1) given by pairs (X_i, V_i), where $1 \leq i \leq N$.
output: the set U of all solutions to (1) in variables $X(N)$.

1. (initialization) Set $a \leftarrow$ the first(the biggest according to $\leq$) vector in V_1 and $k \leftarrow 1$.
2. (extend a and increase k) Set $b \leftarrow$ the first vector in V_k that can glued with a. Set $a \leftarrow a \circ b$ and $k \leftarrow k+1$. If $k = N$ then return (a) and go to step 3 else go to step 2.
3. (modify a) Set $c \leftarrow a_{X_k}$ and $a \leftarrow a_{X(k-1)}$. Set $b \leftarrow$ the first vector $< c$ in V_k that can be glued with a. Then set $a \leftarrow a \circ b$ and go to step 2.
4. (reduce k) Set $k \leftarrow k-1$. If $k = 0$ then stop else go to step 3.

We stress here that if at steps 2 and 3 the claimed vector b doesn't exist the Algorithm goes straight to steps 3 and 4 respectively. The Algorithm obviously solves the problem and it passes through every $a \in U_k$ at most q^l times for every k. The figure q^l may be reduced via a proper ordering of the elements of the sets V_k and this doesn't change the asymptotic running time. The asymptotic running time is as that of the Gluing Algorithm.

3 Complexity Analysis

Lemma 1 *Let $Y \subseteq X$ denote two sets of variables, V be a fixed set of X-vectors and U be the set of Y-vectors, solutions to a randomly chosen equation $f(Y) = 0$ in variables Y. Let*

$$(X, V') = (X, V) \circ (Y, U),$$

where the size $|V'|$ of V' is a random variable. Then for its expectation we have

$$E|V'| = |V|/q.$$

Proof Let V_u denote the subset of vectors from V the Y-subvector of which is u. Then

$$|V| = \sum_u |V_u|$$

and

$$|V'| = \sum_u |V_u| I_u,$$

where $I_u = 0$ if u doesn't occur in U and $I_u = 1$ otherwise. One sees that

$$Pr(I_u = 1) = 1/q,$$

because f is a random equation in Y and is uniquely defined by a mapping of all possible Y-vectors over F_q to F_q. Therefore,

$$E|V'| = \sum_u |V_u| E I_u = 1/q \sum_u |V_u| = |V|/q.$$

This finishes the proof.

Theorem 1 *Let ϵ be any positive real number and $l \geq 3$ and $q \geq 2$ be fixed natural numbers when n tends to infinity. Then the mathematical expectation of the complexity of the Gluing Algorithm is*

$$O((qe^{\gamma_0} + \epsilon)^n + \textit{poly}(n)N)$$

operations, where $\gamma_0 = -\frac{\ln q}{l} - (q^{\frac{1}{l}} - 1)\ln(\frac{1-q^{-1}}{1-q^{-\frac{1}{l}}})$ and poly(n) is a polynomial in n.

Proof By U_k we denote the set of solutions to the subsystem

$$\begin{cases} f_1(X_1) = 0, \\ \dots \\ f_k(X_k) = 0 \end{cases}$$

in $X(k)$-vectors, where $X(k) = X_1 \cup \ldots \cup X_k$. Then

$$(X(k+1), U_{k+1}) = (X(k), U_k) \circ (X_{k+1}, V_{k+1}),$$

where $1 \leq k \leq N-1$. Let ξ_k be the size of the set U_k. One sees that the complexity of the Gluing Algorithm is bounded by $O(\xi + \text{poly}(n)N)$, where $\xi = \text{poly}(n)(\xi_1 + \ldots + \xi_N)$. This is an upper bound on the cost of $N-1$ gluings. It is obvious $\xi_1, \ldots, \xi_N$ are random variables. One finds

$$E\xi = \text{poly}(n) \sum_{k=1}^{N} E\xi_k = \text{poly}(n) max_k E\xi_k$$

because $E\xi_k = O(q^n/q^k)$ by Lemma 1. We now estimate $E\xi_k$.

Because $X_1, \ldots, X_k$ are independent random subsets of X of size $l_1, \ldots, l_k$, the size of the set $X(k) = X_1 \cup \ldots \cup X_k$ is the random variable $\nu(k) = \nu(l_1, \ldots, l_k; n)$ studied in Section 5. Lemma 1 implies

$$E\xi_k = Eq^{\nu(k)-k}.$$

By Theorem 3, Section 5

$$Eq^{\nu(k)-k} \leq \eta_l^n Eq^{\nu_{L_k}-k},$$

where $L_k = l_1 + \ldots + l_k$ and η_l tends to 1 as n tends to infinity since

$$\eta_l = (\prod_{i=1}^{k} \frac{(n+1)\ldots(n+l_i-1)}{(n-1)\ldots(n-l_i+1)})^{1/n}$$

and l_i are bounded by l. By Theorem 4 , Section 5 the value $\eta_l^n Eq^{\nu_{L_k}-k}$ is bounded by

$$O(\eta_l^n (qe^{f(z_\alpha)-\alpha+\alpha\ln\alpha-\frac{\alpha\ln q}{l}})^n)$$

for $\alpha = L_k/n$ and because $k \geq \alpha n/l$. See Section 5 for the definition of $f(z)$ and z_α. Then

$$E\xi = O(\text{poly}(n)\eta_l^n (qe^{\gamma_0})^n),$$

where $\gamma_0 = g(\alpha_0)$ is the maximum of $g(\alpha) = f(z_\alpha) - \alpha + \alpha\ln\alpha - \frac{\alpha\ln q}{l}$ for $\alpha > 0$.

By Lemma 2, proved below, we put

$$\gamma_0 = -\frac{\ln q}{l} - (q^{\frac{1}{l}} - 1)\ln(\frac{1-q^{-1}}{1-q^{-\frac{1}{l}}}).$$

It is easy to see that

$$\text{poly}(n)\eta_l^n (qe^{\gamma_0})^n = O((qe^{\gamma_0} + \epsilon)^n)$$

for any positive real number ϵ and n tending to infinity. This finishes the proof of the Theorem.

Lemma 2 *Let $q \geq 2$ and $l \geq 3$ be natural numbers. Then for $\alpha > 0$ the function $g(\alpha)$ has just one maximum value*

$$g(\alpha_0) = -\frac{\ln q}{l} - (q^{\frac{1}{l}} - 1)\ln(\frac{1-q^{-1}}{1-q^{-\frac{1}{l}}})$$

and $\alpha_0 < l/2$.

Proof We know that $z = z_\alpha$ satisfies

$$\frac{ze^z}{e^z + q^{-1} - 1} = \alpha \tag{4}$$

and $\alpha > 0$ if and only if $z > 0$. We consider $g(\alpha)$ as a function in z, when $z > 0$. In order to find its extremum we take the first derivative. One sees

$$\frac{\partial g}{\partial z} = -\frac{\partial \alpha}{\partial z}(\frac{\ln q}{l} - \ln(\frac{\alpha}{z}))$$

One also sees that $\frac{\partial \alpha}{\partial z}$ only has positive values when $z > 0$. So $\frac{\partial g}{\partial z} = 0$ if and only if $\alpha/z = q^{\frac{1}{l}}$ and so $\alpha_0 = q^{\frac{1}{l}} z_{\alpha_0}$. It follows from (13) that

$$e^{z_{\alpha_0}} = \frac{1 - q^{-1}}{1 - q^{-\frac{1}{l}}}$$

and it is straightforward to see that

$$g(\alpha_0) = -\frac{\ln q}{l} - (q^{\frac{1}{l}} - 1)\ln(\frac{1 - q^{-1}}{1 - q^{-\frac{1}{l}}}).$$

Then

$$e^{z_{\alpha_0}} = 1 + 1/q^{\frac{1}{l}} + \ldots + 1/q^{\frac{l-1}{l}} \leq l.$$

So $z_{\alpha_0} \leq \ln l$ and $\alpha_0 \leq \ln l + 1$, see Lemma 3 in Section 5. This implies $\alpha_0 < l/2$ for $l \geq 6$. But the last statement of the Lemma is also true for $l = 3, 4, 5$ as one sees from taking the maximum of

$$\alpha_0 = q^{\frac{1}{l}} \ln(\frac{1 - q^{-1}}{1 - q^{-\frac{1}{l}}})$$

in q for $q \geq 2$. This proves the Lemma.

4 Even faster Gluing

We'll explain here a variant of the Gluing Algorithm working faster than that introduced in Section 2. Let for the simplicity $N \geq n$ and $l_1 = \ldots = l_N = l$. Then Lemma 2 implies that there exists just one real number $\alpha_1 > 0$ such that

$$g(\alpha_1) = g(2\alpha_1)$$

and $\alpha_1 < \alpha_0 < 2\alpha_1 \leq l$. So one finds natural numbers k_1 and k_2 such that

$$\frac{(k_1 - 1)l}{n} < \alpha_1 \leq \frac{k_1 l}{n}$$

and

$$\frac{(k_1 + k_2 - 1)l}{n} < 2\alpha_1 \leq \frac{(k_1 + k_2)l}{n}.$$

Then the two subsystems of equations (1) are considered:

$$\begin{cases} f_1(X_1) = 0, \\ \ldots \\ f_{k_1}(X_{k_1}) = 0 \end{cases}$$

and

$$\begin{cases} f_{k_1+1}(X_{k_1+1}) = 0, \\ \dots \\ f_{k_1+k_2}(X_{k_1+k_2}) = 0. \end{cases}$$

Let $X' = X_1 \cup \ldots \cup X_{k_1}$ and V' be the set of all solutions to the first subsystem in X'-vectors. Similarly, let $X'' = X_{k_1+1} \cup \ldots \cup X_{k_1+k_2}$ and V'' be the set of all solutions to the second subsystem in X''-vectors.

Gluing2 Algorithm
input: the system of equations (1) given by pairs (X_i, V_i), where $1 \leq i \leq N$.
output: the set U of all solutions to (1) in variables $X_1 \cup \ldots \cup X_N$.
apply the Gluing Algorithm to find (X', V') and (X'', V'')
set $(Z, U) \leftarrow (X', V') \circ (X'', V'')$ and $k \leftarrow k_1 + k_2 + 1$
while $k \leq N$ do
$(Z, U) \leftarrow (Z, U) \circ (X_k, V_k)$ and $k \leftarrow k + 1$,
return (Z, U).

Theorem 2 *Let ϵ be any positive real number and $l \geq 3$ and $q \geq 2$ be fixed natural numbers, when n tends to infinity. Then the mathematical expectation of the complexity of the Gluing2 Algorithm is*

$$O((qe^{g(\alpha_1)} + \epsilon)^n + \textit{poly}(n)N) \tag{5}$$

operations.

The proof follows from the facts that the complexity of finding (X', V'), (X'', V'') and $(X(k_1 + k_2), U_{k_1+k_2})$ is bounded by (5), see formula (3), and the function $g(\alpha)$ has only one maximum at α_0, where $\alpha_1 < \alpha_0 < 2\alpha_1$.

The bound on the complexity of the above algorithm is always better than that on the complexity of the Gluing Algorithm because $g(\alpha_1) < g(\alpha_0)$. The drawback of the Gluing2 Algorithm is that the amount of memory used is of the same order of magnitude as the running time.

5 Random allocations by complexes

In this section we use some results and ideas from the Random Allocations Theory, see [7], in order to prove Theorem 4 which is basic for the complexity analysis of the Gluing Algorithm. Let there be n boxes into which k complexes of particles are independently thrown, $l_i \leq n$ particles at the i-th throw. This means that at the i-th throw $\binom{n}{l_i}$ boxes are occupied with the equal probability $1/\binom{n}{l_i}$. We introduce two random variables now. Let

$$\nu = \nu(k) = \nu(l_1, \ldots, l_k; n)$$

be the number of filled boxes after k such throws and ν_L be the number of filled boxes when L particles are thrown independently of each other. One sees that

$$\nu_L = \nu(1, \ldots, 1; n),$$

where L is the number of 1's in the last formula. In this Section we'll prove some auxiliary statements on the distribution of the random variables x^ν and x^{ν_L}, where x is first a variable and then a real number ≥ 1. By Ex^ν and Ex^{ν_L} we denote mathematical expectations of these random variables.

Theorem 3 *1. Let x be any variable then*

$$Ex^{\nu} = \sum_{m=0}^{n} \frac{\binom{m}{l_1}\dots\binom{m}{l_k}}{\binom{n}{l_1}\dots\binom{n}{l_k}} \binom{n}{m} (1-x)^{n-m} x^m.$$

2. Let x be a real number ≥ 1 and $L = l_1 + \ldots + l_k$ then

$$Ex^{\nu_L} \leq Ex^{\nu} \leq \prod_{i=1}^{k} \frac{(n+1)\dots(n+l_i-1)}{(n-1)\dots(n-l_i+1)} Ex^{\nu_L}.$$

Proof In order to prove the first statement we divide the boxes into two groups of n_1 and $n-n_1$ respectively. Let t be an integer number and $0 \leq t \leq n$. We now calculate the probability of the event $\nu = t$ by considering the joint probability, fixing the number of particles s_i and $l_i - s_i$ of the i-th complex in the groups obtained. One sees

$$Pr(\nu = t) = \sum_{i,s_i=0}^{l_i} \frac{\binom{n_1}{s_1}\dots\binom{n_1}{s_k}\binom{n-n_1}{l_1-s_1}\dots\binom{n-n_k}{l_k-s_k}}{\binom{n}{l_1}\dots\binom{n}{l_k}} \times$$

$$\sum_{t_1=0}^{t} Pr(\nu(s_1,\dots,s_k;n_1) = t_1)\ Pr(\nu(l_1-s_1,\dots,l_k-s_k;n-n_1) = t-t_1), \qquad (6)$$

where we run over all $1 \leq i \leq k$ and $0 \leq s_i \leq l_i$ in the first sum. We multiply the both sides of (6) by x^t and sum over t from 0 to n.We denote

$$E(l_1,\dots,l_k;n;x) = \binom{n}{l_1}\dots\binom{n}{l_k} Ex^{\nu(l_1,\dots,l_k;n)}.$$

Then (6) implies

$$E(l_1,\dots,l_k;n;x) =$$

$$\sum_{i,s_i=0}^{l_i} E(s_1,\dots,s_k;n_1;x)\ E(l_1-s_1,\dots,l_k-s_k;n-n_1;x). \qquad (7)$$

We multiply the both sides of the last formula by $z_1^{l_1}\dots z_k^{l_k}$, where $z_1,\dots,z_k$ are any variables, and sum over all l_i from 0 to n. Denoting

$$F(n;x;z_1,\dots,z_k) = \sum_{i,l_i=0}^{n} z_1^{l_1}\dots z_k^{l_k} E(l_1,\dots,l_k;n;x), \qquad (8)$$

we get from (7) that

$$F(n;x;z_1,\dots,z_k) = F(n_1;x;z_1,\dots,z_k)F(n-n_1;x;z_1,\dots,z_k) = F(1;x;z_1,\dots,z_k)^n. \qquad (9)$$

One easily sees that

$$E(l_1,\dots,l_k;1;x) = \begin{cases} 1, & \text{if } l_1 = \dots l_k = 0, \\ x, & \text{otherwise.} \end{cases}$$

Then

$$F(1;x;z_1,\dots,z_k) = 1 - x + x(1+z_1)\dots(1+z_k)$$

and

$$F(n;x;z_1,\dots,z_k)=(1-x+x(1+z_1)\dots(1+z_k))^n=$$
$$=\sum_{m=0}^{n}\binom{n}{m}((1+z_1)\dots(1+z_k))^m(1-x)^{n-m}x^m.$$

From

$$((1+z_1)\dots(1+z_k))^m=\sum_{i,l_i=0}^{m}\binom{m}{l_1}\dots\binom{m}{l_k}z_1^{l_1}\dots z_k^{l_k},$$

it follows that

$$F(n;x;z_1,\dots,z_k)=$$
$$\sum_{i,l_i=0}^{n}z_1^{l_1}\dots z_k^{l_k}(\sum_{m=0}^{n}\binom{m}{l_1}\dots\binom{m}{l_k}\binom{n}{m}(1-x)^{n-m}x^m).$$

because $\binom{m}{l_i}=0$ for $m<l_i$. This with (8) together imply

$$E(l_1,\dots,l_k;n;x)=\binom{n}{l_1}\dots\binom{n}{l_k}Ex^\nu=$$
$$\sum_{m=0}^{n}\binom{m}{l_1}\dots\binom{m}{l_k}\binom{n}{m}(1-x)^{n-m}x^m$$

and proves the first statement of the Theorem.

Let us prove the second statement. From the first statement we see that

$$Ex^\nu=$$
$$\frac{n^L}{\binom{n}{l_1}\dots\binom{n}{l_k}}\sum_{m=0}^{n}\frac{\binom{m}{l_1}\dots\binom{m}{l_k}}{n^L}\binom{n}{m}(1-x)^{n-m}x^m.$$

Then

$$\frac{\binom{m}{l_1}\dots\binom{m}{l_k}}{n^L}=\prod_{i=1}^{k}\frac{m}{n}(\frac{m}{n}-\frac{1}{n})\dots(\frac{m}{n}-\frac{l_i-1}{n})=$$
$$\sum_{s=0}^{L}d_s(\frac{m}{n})^{L-s}$$

is a polynomial in $\frac{m}{n}$ with some rational coefficients $d_0,d_1,\dots,d_L$. So

$$Ex^\nu=$$
$$\frac{n^L}{\binom{n}{l_1}\dots\binom{n}{l_k}}\sum_{s=0}^{L}d_s\sum_{m=0}^{n}(\frac{m}{n})^{L-s}\binom{n}{m}(1-x)^{n-m}x^m=$$
$$\frac{n^L}{\binom{n}{l_1}\dots\binom{n}{l_k}}\sum_{s=0}^{L}d_sEx^{\nu_{L-s}}$$

because the first statement of the Theorem implies

$$Ex^{\nu_s}=\sum_{m=0}^{n}(\frac{m}{n})^s\binom{n}{m}(1-x)^{n-m}x^m.$$

Therefore,

$$Ex^{\nu} \leq \frac{n^L}{\binom{n}{l_1}\dots\binom{n}{l_k}}(\sum_{s=0}^{L}|d_s|)Ex^{\nu_L}, \tag{10}$$

because of $Ex^{\nu_s} \leq Ex^{\nu_L}$, when $0 \leq s \leq L$ and due to $x \geq 1$. Now

$$\sum_{s=0}^{L}|d_s| \leq \prod_{i=1}^{k}(1+\frac{1}{n})\dots(1+\frac{l_i-1}{n}).$$

Really, $|d_s|$ being the coefficient of the polynomial

$$\prod_{i=1}^{k}x(x-\frac{1}{n})\dots(x-\frac{l_i-1}{n})$$

at x^{L-s}, is bounded by the coefficient of the polynomial

$$\prod_{i=1}^{k}x(x+\frac{1}{n})\dots(x+\frac{l_i-1}{n})$$

at x^{L-s}. So the sum $\sum_{s=0}^{L}|d_s|$ is bounded by the value of the last polynomial at $x=1$. Hence, (10)

$$Ex^{\nu_L} \leq Ex^{\nu} \leq \prod_{i=1}^{k}\frac{(1+1/n)\dots(1+(l_i-1)/n)}{(1-1/n)\dots(1-(l_i-1)/n)}Ex^{\nu_L},$$

because $Ex^{\nu_L} \leq Ex^{\nu}$ is obvious. This implies the Theorem.

Remark 1 *The formula for Ex^{ν} given in Theorem 3 is not convenient to compute with the float point arithmetic, because in this alternating series some summands are much bigger than the final result. One can instead use the following recurrent formula for computing probabilities $Pr(\mu=t)$:*

$$Pr(\mu(l_1+1,l_2\dots,l_k;n)=t) =$$
$$Pr(\mu(l_1,l_2\dots,l_k;n)=t)\frac{n-t-l_1}{n-l_1} + Pr(\mu(l_1,l_2\dots,l_k;n)=t+1)\frac{t+1}{n-l_1},$$

where $\mu = n-\nu$ is the number of empty boxes. This is a simple generalization of the formula (5) at page 3 in [7]. Then the expectation of x^{ν} is computed using the definition:

$$Ex^{\nu} = \sum_{t=0}^{n}Pr(\mu(l_1,l_2\dots,l_k;n)=t)x^{n-t}.$$

Let

$$f(z) = \ln(e^z + q^{-1} - 1) - \alpha\ln(z)$$

be a real-valued function in the real-valued variable z for a positive number α. By z_{α} we denote the only positive root of the equation

$$\frac{\partial f}{\partial z}(z) = 0 \tag{11}$$

in z. One easily proves that there is only one such a root, see Lemma 3.

Theorem 4 *Let δ be a fixed positive number such that $0 < \delta < 1$. Then for any real $q \geq 1$*

$$Eq^{\nu_L} = \begin{cases} < q^{n^\delta}, & \text{if } L < n^\delta; \\ O((qe^{f(z_\alpha)-\alpha+\alpha\ln\alpha})^n), & \text{if } L \geq n^\delta, \end{cases}$$

where $\alpha = L/n$ and n tends to infinity.

Proof The statement is trivial for $L < n^\delta$. Let us prove this for $L \geq n^\delta$. We have

$$Eq^{\nu_L} = q^n E(1/q)^{\mu_L},$$

where $\mu_L = n - \nu_L$ is the number of empty boxes, when L particles are independently thrown in n boxes. Then formulas (25) and (29) of [7] imply

$$E(1/q)^{\mu_L} = \frac{L!}{2\pi i n^L} \oint (e^z + q^{-1} - 1)^n \frac{dz}{z^{L+1}},$$

where the integral is over any closed contour encircling the point $z = 0$. A more general integral was estimated in [3] following Theorem 3 of [6], so we get

$$E(1/q)^{\mu_L} = \frac{L!}{n^L} \frac{e^{nf(z_\alpha)}}{z_\alpha \sqrt{2\pi n \frac{\partial^2 f}{\partial z^2}(z_\alpha)}} (1 + O(\frac{1}{z_\alpha n})) \tag{12}$$

as n tends to infinity. We see that

$$\frac{\partial f}{\partial z} = \frac{e^z}{e^z + q^{-1} - 1} - \frac{\alpha}{z}.$$

So

$$\frac{e^{z_\alpha}}{e^{z_\alpha} + q^{-1} - 1} = \frac{\alpha}{z_\alpha}. \tag{13}$$

We'll prove a technical Lemma now.

Lemma 3 *1. For any positive real number α there exists only one solution z_α to the equation (11) and*

$$\max(\alpha/q, \alpha - 1) \leq z_\alpha \leq \alpha,$$

2. there is a constant c such that

$$1 \leq \frac{\alpha}{z_\alpha^2 \frac{\partial^2 f}{\partial z^2}(z_\alpha)} \leq c$$

for any positive number α.

Proof The function $ze^z/(e^z + q^{-1} - 1)$ has a positive derivative for a positive z, so for any $\alpha > 0$ there is only one z_α satisfying (13). Having represented (13) as

$$\frac{1}{1 - (1 - q^{-1})/e^{z_\alpha}} = \frac{\alpha}{z_\alpha}$$

one sees that $\alpha/z_\alpha \leq q$, due to $q \geq 1$. We also realize that

$$z_\alpha = \alpha + w(-\frac{\alpha(1 - q^{-1})}{e^\alpha}),$$

where $w = w(-\frac{\alpha(1-q^{-1})}{e^\alpha})$ is the closest to 0 solution of the equation

$$we^w = -\frac{\alpha(1 - q^{-1})}{e^\alpha}.$$

It is easy to prove that $-1 \leq w \leq 0$, so $\alpha \geq z_\alpha \geq \alpha - 1$. This proves the first statement of the Lemma.

To prove the second statement one finds

$$\frac{\partial^2 f}{\partial z^2} = \frac{e^z(q^{-1}-1)}{(e^z+q^{-1}-1)^2} + \frac{\alpha}{z^2}.$$

Therefore,

$$\beta = \frac{\alpha}{z_\alpha^2 \frac{\partial^2 f}{\partial z^2}(z_\alpha)} = \frac{1}{1-(1-q^{-1})\frac{z_\alpha}{(e^{z_\alpha}+q^{-1}-1)}}.$$

So $1 \leq \beta < c$ for some constant c and any $z_\alpha > 0$, and so for any $\alpha > 0$. This proves the Lemma on the whole.

Let us return to the proof of the Theorem. Using Stirling approximation for the factorial function, one gets from (12) that

$$E(1/q)^{\mu_L} = \alpha^{\alpha n} e^{n(f(z_\alpha)-\alpha)} \sqrt{\frac{\alpha}{z_\alpha^2 \frac{\partial^2 f}{\partial z^2}(z_\alpha)}} (1 + O(1/L + 1/(nz_\alpha)).$$

We have

$$nz_\alpha \geq \alpha n/q = L/q \geq n^\delta/q$$

because of $L \geq n^\delta$. So

$$1/L + 1/(nz_\alpha) = o(1)$$

as n tends to infinity. Using statement 2. of Lemma 3 we get $E(1/q)^{\mu_L} = O(e^{n(f(z_\alpha)-\alpha+\alpha\ln\alpha)})$. This proves the Theorem.

References

[1] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, in Eurocrypt 2000, LNCS **1807**, Springer, 392-407.

[2] N. Courtois, J. Pieprzyk, *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*, in Asiacrypt 2002, LNCS **2501**, Springer, 267-287.

[3] V.P.Chistyakov, *Discrete limit distributions in the problem of shots with arbitrary probabilities of occupancy of boxes*, Matem.Zametki, **1**, 1(1967), 9-16.

[4] J.-C. Faugère, A. Joux, *Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases*, in Crypto 2003, LNCS **2729**, Springer, 44-60.

[5] K. Iwama, *Worst-Case Upper Bounds for kSAT*, The Bulletin of the EATCS, (82), 2004, pp.61-71.

[6] V.Kolchin, *The rate of convergence to limit distributions in the classical problem of shots*, Teoriya veroyatn. i yeye primenen., **11**, 1(1966), 144-156.

[7] V.Kolchin, A.Sevast'yanov, and V.Chistyakov, *Random allocations*, John Wiley & Sons, 1978.

[8] H.Raddum, *Solving non-linear sparse equation systems over $GF(2)$ using graphs*, preprint, 2004.

[9] H.Raddum, I.Semaev, the article in preparation.