Polynomial time factoring algorithm using Bayesian arithmetic

Michel Feldmann *

Abstract

In a previous paper, we have shown that any Boolean formula can be encoded as a linear programming problem in the framework of Bayesian probability theory. When applied to NP-complete algorithms, this leads to the fundamental conclusion that P = NP. Now, we implement this concept in elementary arithmetic and especially in multiplication. This provides a polynomial time deterministic factoring algorithm, while no such algorithm is known to day. This result clearly appeals for a revaluation of the current cryptosystems. The Bayesian arithmetic environment can also be regarded as a toy model for quantum mechanics.

1 Introduction

Arithmetic is a part of abstract mathematics, i.e., a theory based, for instance, on Peano axioms and dealing with infinitely many elements. On the other hand, arithmetic is also a practical way of counting and computing. Paradoxically, computation rules are not very much concerned by the abstract theory. Instead, only Boolean operations are at work, e.g., in micro-processors. In a previous paper [1], we have shown that Boolean operations can be described in terms of Bayesian probability leading to the stunning result P = NP. We propose in the present paper to investigate the Bayesian structure of a particular environment, namely, elementary arithmetic operations.

In this approach, we start with a Boolean algebra composed of all relevant binary digits involved in a Diophantine equation. In order to deal with tractable formulae, it is suitable to introduce internal variables beforehand, e.g. carry bits. Now, we construct a Bayesian environment and we account for the rules of arithmetic by means of structural equations. These constraints will be further added to a number of specific equations corresponding to the input data and a number of universal equations reflecting the laws of logics. The Bayesian method consists in checking the consistency of all these conditions by linear programming (LP). When feasible, the undefined bits are eventually computed.

The model is founded on the theory of probability. Nevertheless, when the LP problem is feasible in the *general environment*, the system always accept strictly deterministic solutions. This is easily proved [1] by exploring the full ensemble of possible assignments. By contrast, in the present *arithmetic environment*, the use of internal variables impose a limitation in the set of accessible assignments, because internal variables cannot be assigned independently. Therefore only a part of the potential assignments can be consistently explored and the proof is no longer valid. As a result, a feasible LP problem may or not accept deterministic solutions. For instance, in the factoring algorithm, the LP problem is generally feasible: When the input integer is composite, its factors are derived from the deterministic solutions. On the contrary, when the input is prime, we have no deterministic non-trivial factors but we do have probabilistic solutions. In this respect, the Bayesian system can be regarded as a toy model of quantum formalism: a 'composite system' is likened to a classical object with deterministic parameters while a 'prime system' is likened to a quantum object with only probabilistic parameters. We will shortly sketch an example in Sec. 3.3. Fortunately, LP is quite efficient to compute the existing deterministic solutions or decide with certainty that no such solution exists. Thus, we obtain both a deterministic polynomial time factoring algorithm and a deterministic polynomial time criterion of primality. Presently, only a quantum algorithm is known for the first case [2] and a class-P algorithm was only found recently [3] for the second case.

The general framework of the theory is the following: We consider a Boolean binary algebra with N variables X_i, for $i \in [\![1, N]\!]$. Thus, we may potentially assign a value 0 or 1 to each variable. We name *complete assignment* a full assignment to the N variables and *partial assignment* an assignment to less than N variables. We note $\overline{\mathsf{X}}_i$ the negation of X_i, and call *literal* a variable or its negation. Given two logical formulae (or *decision functions*) f_1 and f_2, it is convenient to note $(\mathsf{f}_1;\mathsf{f}_2)$ (with a semicolon) the conjunction $\mathsf{f}_1 \wedge \mathsf{f}_2$ and $(\mathsf{f}_1,\mathsf{f}_2)$ (with a comma) the disjunction $\mathsf{f}_1 \vee \mathsf{f}_2$. We name *requirement* a conjunction of literals, *complete requirement* a conjunction of N literals, e.g., $\Xi = (\mathsf{X}_1;\overline{\mathsf{X}}_2;\ldots;\mathsf{X}_N)$, that is satisfiable by a complete assignment, e.g., $\xi = (1, 0, \ldots, 1)$, and *partial requirement* a conjunction of less than N literals, e.g., $(\mathsf{X}_i;\overline{\mathsf{X}}_j;\mathsf{X}_k)$. Clearly, there are 2^N different complete assignments and therefore 2^N complete requirements. On the other hand, with up to N variables, it is possible to construct 2^{2^N} different decision functions.

*Electronic address: michel.feldmann@polytechnique.org

Now, we propose to regard any *decision function* as a *random event* and to reformulate the logical equations as a set of linear equations between the *probabilities* of the relevant requirements. For this, we use the Bayesian conception of the theory of probability [4]. Given by hypothesis that a particular logical proposition (Λ) has to be satisfied, *the probability of any event will be conditioned by* (Λ). For instance, in the conventional addition of two integers U and V, (Λ) will be the statement (Σ) that the two integers U and V sum to a third integer S.

The basic probability set is the ensemble $\Omega = \{\Xi\}$ of all 2^N complete requirements, labelled by the 2^N complete assignments ξ. Since the cardinality of Ω is finite, the power set $\mathcal{P}(\Omega)$, of cardinality 2^{2^N}, is a sigma-algebra $\mathcal{T}$, identical to the ensemble of all decision functions. Now we have to define a probability distribution P on $\mathcal{T}$ conditioned by (Λ). Finally, the Kolmogorov probability space is $(\Omega, \mathcal{T}, \mathrm{P})$.

We start with the prior information that (Λ) is TRUE and determine how this knowledge affects the conditional probability of the relevant requirements. It turns out that these constraints are conveniently formulated as a LP problem. Therefore, we complete the computation by solving this LP problem. For NP-algorithms, the number of relevant requirements scales as $\mathrm{O}(N^K)$, where K is an integer. Thus, the LP solutions are obtained in polynomial time.

In this paper, we will investigate the behaviour of elementary arithmetic operations in such a Bayesian context. We will first implement with full details the *Bayesian addition*. Needless to say that the method is completely maladjusted for practical operations and cannot compete with a direct computation. However the derivation is quite simple and adequate to clarify the present concept. Furthermore, Bayesian addition is a part of *Bayesian multiplication*: Again, the method is by far too complicated when compared with a direct product but the interest lies in the inverse problem, namely factorization. Even for this last problem, the method, at least in its present state, is complicated for small numbers. But the unique feature is clearly the scaling capability: Factorization of an integer of n bits is obtained by LP in a system of $\mathrm{O}(n^2)$ unknowns. This remains the only possibility to factorize integers of hundreds or even thousands of bits.

2 Addition of two integers

Let U and V be two integers. Without loss of generality we can assume that they are both described by the same number n of bits, given that we may complete by a number of zeros if necessary. Let $S = U + V$ be the sum. The binary expansions read,

$$U = \sum_{i=0}^{n-1} u_i 2^i \ ; \ V = \sum_{i=0}^{n-1} v_i 2^i \ ; \ S = U + V = \sum_{i=0}^{n} s_i 2^i$$

with $u_i, v_i, s_i \in \{0, 1\}$. It is suitable to introduce carry bits explicitly.

Example: Let $n = 2$ and let r_i be the carry bits. The binary operation can be written as

U		u_1	u_0
V		v_1	v_0
R	r_2	r_1	.
S	s_2	s_1	s_0

In Bayesian arithmetic, all bits, including the carry bits are considered as random variables. The assignments u_i, v_i, s_i and r_i are likened to the outcomes of these random variables, respectively U_i, V_i, S_i and R_i.

Let Σ be the logical proposition: 'S is the sum of U and V'. *We will compute the conditional probabilities of all events given* Σ.

We have first to define the input data. Provisionally, we suppose that U and V are given, but it would be possible to choose different inputs, e.g., S and U or even exotic data like carry bits. Then, for $i \in [\![0, n-1]\!]$ we suppose that $\mathsf{U}_i = u_i$ and $\mathsf{V}_i = v_i$ with certainty, i.e., with a probability 1. The probabilistic formulation is

$$\mathrm{P}(\mathsf{U}_i = u_i | \Sigma) = 1 \tag{1}$$

$$\mathrm{P}(\mathsf{V}_i = v_i | \Sigma) = 1. \tag{2}$$

The partial probabilities, e.g., $\mathrm{P}(\mathsf{U}_i = u_i|\Sigma)$ are regarded as the *unknowns* of the problem. More generally, we will call *partial probability* the probability of any requirement, conditioned in this section by Σ, and identify such partial probabilities with unknowns (not to be mistaken for the very *variables*, like e.g. U_i). This codification define a set of $2n$ linear equations that we will call *data specific equations*. We will next define *structural equations*, expressing the rules of arithmetic, and later *universal equations*, expressing the laws of logics.

In order to construct the structural equations, let us consider the following *one-bit full adder truth table*.

U_i	V_i	R_i	S_i	R_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

For $i = 0$, the outcome of the variable R_0 is always zero with certainty and therefore this variable will be omitted. For $i = n$ we have

$$\mathrm{P}(\mathsf{S}_n = 1|\Sigma) = \mathrm{P}(\mathsf{R}_n = 1|\Sigma) \tag{3}$$

Therefore, we will not discriminate between S_n and R_n. It will be convenient to keep rather R_n and simply omit the variable S_n.

Example: Let $n = 2$. We will deal with the following variables: U_0, U_1, V_0, V_1, S_0, S_1, R_1, R_2. The truth tables for $i = 1, 0$ read respectively

U_1	V_1	R_1	S_1	R_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

U_0	V_0	S_0	R_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Coming back to the general case, we undertake to translate the truth tables into linear equations between partial probabilities. This is straightforward because the probability of an union of mutually exclusive events is the sum of the probability of each event.

For instance, we read that $(\mathsf{S}_i = 1)$ is the union of four mutually exclusive conjunctions, namely, $[(\mathsf{U}_i = 1) \wedge (\mathsf{V}_i = 1) \wedge (\mathsf{R}_i = 1)]$, $[(\mathsf{U}_i = 1) \wedge (\mathsf{V}_i = 0) \wedge (\mathsf{R}_i = 0)]$, $[(\mathsf{U}_i = 0) \wedge (\mathsf{V}_i = 1) \wedge (\mathsf{R}_i = 0)]$ and $[(\mathsf{U}_i = 0) \wedge (\mathsf{V}_i = 0) \wedge (\mathsf{R}_i = 1)]$. Therefore, the probability of $(\mathsf{S}_i = 1)$ is the sum of the probabilities of the four conjunctions. Note that we are not concerned by the event $(\mathsf{S}_i = 0)$ because its probability is *logically* connected with the probability of $(\mathsf{S}_i = 1)$. This will be accounted for later, by the universal equations.

Now we can construct the structural equations of the addition environment by inspection of the truth tables.

- For $i \in [\![1, n-1]\!]$ we obtain $2(n-1)$ equations,

$$\begin{aligned}
\mathrm{P}(\mathsf{S}_i = 1|\Sigma) &= \mathrm{P}(\mathsf{U}_i = 0; \mathsf{V}_i = 0; \mathsf{R}_i = 1|\Sigma) + \mathrm{P}(\mathsf{U}_i = 0; \mathsf{V}_i = 1; \mathsf{R}_i = 0|\Sigma) \\
&\quad + \mathrm{P}(\mathsf{U}_i = 1; \mathsf{V}_i = 0; \mathsf{R}_i = 0|\Sigma) + \mathrm{P}(\mathsf{U}_i = 1; \mathsf{V}_i = 1; \mathsf{R}_i = 1|\Sigma) \\
\mathrm{P}(\mathsf{R}_{i+1} = 1|\Sigma) &= \mathrm{P}(\mathsf{U}_i = 0; \mathsf{V}_i = 1; \mathsf{R}_i = 1|\Sigma) + \mathrm{P}(\mathsf{U}_i = 1; \mathsf{V}_i = 0; \mathsf{R}_i = 1|\Sigma) \\
&\quad + \mathrm{P}(\mathsf{U}_i = 1; \mathsf{V}_i = 1; \mathsf{R}_i = 0|\Sigma) + \mathrm{P}(\mathsf{U}_i = 1; \mathsf{V}_i = 1; \mathsf{R}_i = 1|\Sigma)
\end{aligned}$$

- For $i = 0$, we have two particular equations: Since R_0 is omitted, we have only one or two relevant conjunctions in each equation,

$$\begin{aligned}
\mathrm{P}(\mathsf{S}_0 = 1|\Sigma) &= \mathrm{P}(\mathsf{U}_0 = 0; \mathsf{V}_0 = 1|\Sigma) + \mathrm{P}(\mathsf{U}_0 = 1; \mathsf{V}_0 = 0|\Sigma) \\
\mathrm{P}(\mathsf{R}_1 = 1|\Sigma) &= \mathrm{P}(\mathsf{U}_0 = 1; \mathsf{V}_0 = 1|\Sigma).
\end{aligned}$$

- For $i = n$, since we have omitted S_n Eq.(3) is unnecessary and we have nothing to set down.

We have then completed the construction of the $2n$ *structural equations*. With the $2n$ specific data equations we have a total of $4n$ equations.

On the other hand, we have $4n$ random *variables*, namely U_i for $i \in [\![0, n-1]\!]$, V_i for $i \in [\![0, n-1]\!]$, S_i for $i \in [\![0, n-1]\!]$ and R_i pour $i \in [\![1, n]\!]$. It is suitable to bring together these $4n$ variables in a single list X_k of *global variables*, labelled from $k = 1$ to $k = 4n$. We will adopt throughout the labelling convention of table 1. For instance, we have

$$\mathsf{X}_k \overset{(\mathrm{def})}{=} \mathsf{S}_{k-2n-1} \quad \text{if} \quad k \in [\![2n+1, 3n]\!]$$

Example: Let $n = 2$. The addition can be written in terms of global variables X_k as,

U		X_2	X_1
V		X_4	X_3
R	X_8	X_7	.
S	X_8	X_6	X_5

U_i	V_i	S_i	$\mathsf{S}_n = \mathsf{R}_n$	R_i
$i \in [\![0, n-1]\!]$	$i \in [\![0, n-1]\!]$	$i \in [\![0, n-1]\!]$	$i = n$	$i \in [\![1, n]\!]$
$k = i + 1$	$k = i + n + 1$	$k = i + 2n + 1$	$k = 4n$	$k = i + 3n$
$k \in [\![1, n]\!]$	$k \in [\![n + 1, 2n]\!]$	$k \in [\![2n + 1, 3n]\!]$	$k = 4n$	$k \in [\![3n + 1, 4n]\!]$
X_k	X_k	X_k	X_{4n}	X_k

Table 1: Labelling convention of the global variables X_k corresponding to the variables U_i, V_i , S_i and R_i in the addition of two integers U et V

$$\begin{aligned}
&\mathrm{P}(2n+1) = \mathrm{P}(-1; n+1) + \mathrm{P}(1; -n-1)\\
&\mathrm{P}(2n+j) = \mathrm{P}(-j; -j-n; j+3n-1) + \mathrm{P}(-j; j+n; -j-3n+1) + \mathrm{P}(j; -j-n; -j-3n+1) + \mathrm{P}(j; j+n; j+3n-1)\\
&\mathrm{P}(3n+1) = \mathrm{P}(1; n+1)\\
&\mathrm{P}(3n+j) = \mathrm{P}(-j; j+n; j+3n-1) + \mathrm{P}(j; -j-n; j+3n-1) + \mathrm{P}(j; j+n; -j-3n+1) + \mathrm{P}(j; j+n; j+3n-1)
\end{aligned}$$

Table 2: Structural equations of addition expressed in terms of global variables ($j \in [\![2, n]\!]$). We have a total of $2n$ structural equations.

Thanks to this notation, it is possible to make use of a shortcut: We will simply write (k) to describe the formula $(\mathsf{X}_k = 1|\Sigma)$, omitting both the reference to X and the condition Σ. Similarly, we will note $(-k)$ for $(\mathsf{X}_k = 0|\Sigma)$. Now, we replace $(\mathsf{U}_i = 1|\Sigma)$ by $(i+1)$ and $(\mathsf{U}_i = 0|\Sigma)$ by $(-i-1)$. The formula $(\mathsf{V}_i = 1|\Sigma)$ is replaced by $(i+1+n)$, $(\mathsf{S}_i = 1|\Sigma)$ is replaced by $(i+1+2n)$ and for $i > 0$, $(\mathsf{R}_i = 1|\Sigma)$ is replaced by $(i+3n)$. Finally, the set of structural equations are gathered together in Table 2 (where $j = i + 1$ for simplicity).

To sum up, we have $2n$ specific data equation to specify the two input integers U and V and $2n$ structural equations to describe the arithmetic operation of addition.

Example: Let $n = 2$ and consider the addition $2 + 3$. We have $u_0 = 0$, $u_1 = 1$, $v_0 = 1$ and $v_1 = 1$. The $4n = 8$ equations read:

$$\begin{aligned}
&\mathrm{P}(1) = 0;\ \mathrm{P}(-2) = 0;\ \mathrm{P}(-3) = 0;\ \mathrm{P}(-4) = 0.\\
&\mathrm{P}(5) = \mathrm{P}(1; -3) + \mathrm{P}(-1; 3)\\
&\mathrm{P}(6) = \mathrm{P}(-2; -4; 7) + \mathrm{P}(-2; 4; -7) + \mathrm{P}(2; -4; -7) + \mathrm{P}(2; 4; 7)\\
&\mathrm{P}(7) = \mathrm{P}(1; 3)\\
&\mathrm{P}(8) = \mathrm{P}(-2; 4; 7) + \mathrm{P}(2; -4; 7) + \mathrm{P}(2; 4; -7) + \mathrm{P}(2; 4; 7)
\end{aligned} \tag{4}$$

(We have arbitrary chosen to formulate the data equations with zero right hand side).

Remark: We have presented the conventional addition of two integers U and V. Actually, the Bayesian addition also fits onto operations where the input data are not necessarily the bits of U and V but any set of assignments among the $4n$ variables. Let n_b be this number of data bits. We have then $2n + n_b$ equations,

Equations	number
data	n_b
structural	$2n$
Total	$2n + n_b$

For instance, if $n = 2$ and if we consider the subtraction $5-2$, we have $n_b = 5$ and the data specific equations read

$$\mathrm{P}(1) = 0;\ \mathrm{P}(-2) = 0;\ \mathrm{P}(-5) = 0;\ \mathrm{P}(6) = 0;\ \mathrm{P}(-8) = 0. \tag{5}$$

More generally, we may even specify any set of relevant unknowns, that we are going to define.

Relevant unknowns: The unknowns of the LP problem are primary the partial probabilities involved in the set of specific or structural equations. However, these relevant partial probabilities are also involved in universal equations expressing the rules of logics. For instance the equation $\mathrm{P}(k) = 0$ entails the logical consequence $\mathrm{P}(-k) = 1$. We will name $\mathrm{P}(-k)$ a *variant* of $\mathrm{P}(k)$. Similarly, the use of the relevant unknown $\mathrm{P}(k_1; k_2)$ entails the need to account for the logical consequence $\mathrm{P}(k_1) = \mathrm{P}(k_1; k_2) + \mathrm{P}(k_1; -k_2)$ and we will also name $\mathrm{P}(k_1)$ and $\mathrm{P}(k_1; -k_2)$ *variants* of $\mathrm{P}(k_1; k_2)$. In practice, we list all partial probabilities involved in all specific and structural equations: We obtain the variants by iteration in removing one or several literals or in switching a literal into its negation.

The relevant unknowns are the partial probabilities involved in the specific or structural equations or the variants of these partial probabilities.

In order to list the relevant unknowns, let us define a *positive unknown* as an unknown involving only variables and no negation. For instance, $\mathrm{P}(k_1; k_2; \dots)$ will be called positive if and only if $k_1, k_2, \dots > 0$. Now, let us inspect the different equations:

$\mathrm{P}(k)$
$\mathrm{P}(1;n+1)$; $\mathrm{P}(i+1;i+1+n)$; $\mathrm{P}(i+1;i+3n)$; $\mathrm{P}(i+1+n;i+3n)$
$\mathrm{P}(i+1;i+1+n;i+3n)$

Table 3: Relevant positive unknowns involved in the addition of 2 integers of n bits. ($i \in [\![1, n-1]\!]$; $k \in [\![1, 4n]\!]$). We have a total of $8n-3$ positive unknowns.

- We have $4n$ variables X_k and thus $4n$ positive relevant unknowns $\mathrm{P}(k)$ with one literal and finally $8n$ variants with one literal.

- For the bit $i = 0$, we have introduced $\mathrm{P}(1; n+1)$, i.e., one positive unknown and thus 4 variants with two literals.

- For each bit $i \in [\![1, n-1]\!]$ we have introduced $\mathrm{P}(i+1; i+1+n; i+3n)$, i.e., one positive unknown of three literal, and thus 3 positive variants with two literals, and finally 8 variants of three literals and 3×4 variants of 2 literals.

Collecting these results in Table 3, the numbers of relevant unknowns are the following:

Literals	positive unknowns	unknowns
1	$4n$	$8n$
2	$3n-2$	$12n-8$
3	$n-1$	$8n-8$
Total	$8n-3$	$28n-16$

Example: Let $n = 2$. We have $28n - 16 = 40$ relevant unknowns, namely:

$\mathrm{P}(\pm1$, $\mathrm{P}(\pm2)$, $\mathrm{P}(\pm3)$, $\mathrm{P}(\pm4)$, $\mathrm{P}(\pm5)$, $\mathrm{P}(\pm6)$, $\mathrm{P}(\pm7)$, $\mathrm{P}(\pm8)$.

$i = 0$: $\mathrm{P}(\pm1; \pm3)$,

$i = 1$: $\mathrm{P}(\pm2; \pm4; \pm7)$, $\mathrm{P}(\pm2; \pm4)$, $\mathrm{P}(\pm4; \pm7)$, $\mathrm{P}(\pm2; \pm7)$.

The $8n - 3 = 13$ positive relevant unknowns read:

$\mathrm{P}(1)$, $\mathrm{P}(2)$, $\mathrm{P}(3)$, $\mathrm{P}(4)$, $\mathrm{P}(5)$, $\mathrm{P}(6)$, $\mathrm{P}(7)$, $\mathrm{P}(8)$,

$\mathrm{P}(1;3)$, $\mathrm{P}(4;7)$, $\mathrm{P}(2;7)$, $\mathrm{P}(2;4)$,

$\mathrm{P}(2;4;7)$.

Relevant universal equations We are now going to account for the laws of logic. According to consistency theorems by Richard Cox [5] these laws are expressed in the quantitative rules of probability theory [4]. In the present context, they give rise to a number of consistency constraints that we have called *universal equations*. They are conveniently derived from the list of all relevant positive unknowns.

- For each relevant positive unknown of one literal $\mathrm{P}(k)$ we have one normalization equation:

$$1 = \mathrm{P}(k) + \mathrm{P}(-k) \tag{6}$$

that is a total of $4n$ equations.

- For each relevant positive unknown of two literals $\mathrm{P}(k_1; k_2)$, we have 4 consistency equations

$$\begin{aligned} \mathrm{P}(\pm k_1) &= \mathrm{P}(\pm k_1; k_2) + \mathrm{P}(\pm k_1; -k_2) \\ \mathrm{P}(\pm k_2) &= \mathrm{P}(\pm k_2; k_1) + \mathrm{P}(\pm k_2; -k_1) \end{aligned} \tag{7}$$

that is a total of $4 \times (3n-2) = 12n-8$ equations.

- For each relevant positive unknown of three literals $\mathrm{P}(k_1; k_2; k_3)$, we have $3 \times 4 = 12$ consistency equations, namely

$$\begin{aligned} \mathrm{P}(\pm k_1; \pm k_2) &= \mathrm{P}(\pm k_1; \pm k_2; k_3) + \mathrm{P}(\pm k_1; \pm k_2; -k_3) \\ \mathrm{P}(\pm k_2; \pm k_3) &= \mathrm{P}(\pm k_2; \pm k_3; k_1) + \mathrm{P}(\pm k_2; \pm k_3; -k_1) \\ \mathrm{P}(\pm k_3; \pm k_1) &= \mathrm{P}(\pm k_3; \pm k_1; k_2) + \mathrm{P}(\pm k_3; \pm k_1; -k_2) \end{aligned} \tag{8}$$

that is a total of $12 \times (n-1)$ equations.

Collecting these results, the numbers of relevant universal equations are the following:

literals	positive unknowns	universal equations
1	$4n$	$4n$
2	$3n-2$	$12n-8$
3	$n-1$	$12n-12$
Total	$8n-3$	$28n-20$

Example: Let again $n = 2$. We have
- $4n = 8$ normalization equations for 8 positive literals, namely

$$\begin{aligned} 1 = \mathrm{P}(1) + \mathrm{P}(-1)\ ; 1 = \mathrm{P}(2) + \mathrm{P}(-2)\ ; 1 = \mathrm{P}(3) + \mathrm{P}(-3)\ ; 1 = \mathrm{P}(4) + \mathrm{P}(-4) \\ 1 = \mathrm{P}(5) + \mathrm{P}(-5)\ ; 1 = \mathrm{P}(6) + \mathrm{P}(-6)\ ; 1 = \mathrm{P}(7) + \mathrm{P}(-7)\ ; 1 = \mathrm{P}(8) + \mathrm{P}(-8). \end{aligned} \tag{9}$$

- $4 \times (3n - 2) = 16$ universal equations corresponding to $3n - 2 = 4$ positive unknowns with 2 literals.

$$\begin{aligned} \mathrm{P}(\pm 1) = \mathrm{P}(\pm 1; 3) + \mathrm{P}(\pm 1; -3); \quad \mathrm{P}(\pm 3) = \mathrm{P}(\pm 3; 1) + \mathrm{P}(\pm 3; -1) \\ \mathrm{P}(\pm 2) = \mathrm{P}(\pm 2; 7) + \mathrm{P}(\pm 2; -7); \quad \mathrm{P}(\pm 2) = \mathrm{P}(\pm 2; 4) + \mathrm{P}(\pm 2; -4) \\ \mathrm{P}(\pm 4) = \mathrm{P}(\pm 4; 7) + \mathrm{P}(\pm 4; -7); \quad \mathrm{P}(\pm 4) = \mathrm{P}(\pm 4; 2) + \mathrm{P}(\pm 4; -2) \\ \mathrm{P}(\pm 7) = \mathrm{P}(\pm 7; 4) + \mathrm{P}(\pm 7; -4); \quad \mathrm{P}(\pm 7) = \mathrm{P}(\pm 7; 2) + \mathrm{P}(\pm 7; -2) \end{aligned} \tag{10}$$

- $12n - 12 = 12$ universal equations for the single 3-literal positive unknown,

$$\begin{aligned} \mathrm{P}(\pm 2; \pm 4) = \mathrm{P}(\pm 2; \pm 4; 7) + \mathrm{P}(\pm 2; \pm 4; -7) \\ \mathrm{P}(\pm 4; \pm 7) = \mathrm{P}(\pm 4; \pm 7; 2) + \mathrm{P}(\pm 4; \pm 7; -2) \\ \mathrm{P}(\pm 2; \pm 7) = \mathrm{P}(\pm 2; \pm 7; 4) + \mathrm{P}(\pm 2; \pm 7; -4) \end{aligned} \tag{11}$$

that is a total of $28n - 20 = 36$ relevant universal equations. The total number of equations is then (36 universal equations) + (8 specific equations) = 44 equations.

Linear programming implementation At this point, we have completed the conversion of all Boolean formulae into linear equations operating in a real vector space of the *unknown*-vectors with $N = 28n - 16$ dimensions. Thanks to the set of universal equations, these unknowns can be consistently interpreted as partial probabilities in a Bayesian probability space. Therefore they are non negative. This defines a linear programming problem [6, 7] which can be solved in polynomial time in $N = \mathrm{O}(n)$ [8]. If the number of data bits is n_b, we have a total of $n_b + 30n - 20$ linear equations, e.g., $32n - 20$ for the conventional addition.

Due to the product rule in the probability space we have:

$$\begin{aligned} \mathrm{P}(k_1) = 0 \Rightarrow \mathrm{P}(k_1; k_2) = 0 \Rightarrow \mathrm{P}(k_1; k_2; k_3) = 0. \\ \mathrm{P}(k_1) = 1 \Rightarrow \mathrm{P}(k_1; k_2) = \mathrm{P}(k_2) \\ \mathrm{P}(k_1) = 1 \Rightarrow \mathrm{P}(k_1; k_2; k_3) = \mathrm{P}(k_2; k_3) \end{aligned} \tag{12}$$

It is convenient to take advantage of these relations to simplify the linear system by accounting beforehand for the data equations. As a result, a number of equations of the rough system are cancelled and a number of unknowns become irrelevant.

Generally, only *deterministic solutions* are of interest. Note that deterministic solutions are also *separable* [1], i.e., the probability of any requirement is the product of the probabilities of its literals, e.g.,

$$\mathrm{P}(k_1; k_2; k_3) = \mathrm{P}(k_1) \times \mathrm{P}(k_2) \times \mathrm{P}(k_3). \tag{13}$$

When the LP problem is feasible, we obtain a value for each relevant partial probability. Deterministic solutions are in principle computed by LP-optimization in a time $\mathrm{O}(n)$. However, in this simple case of addition, it turns out that a feasible problem always accept a deterministic solution and even optimization is unnecessary when accounting for Eq.(12). The conclusion will be different for multiplication.

When the LP problem is not feasible, the problem has no solution, e.g., $S - U$ when $S < U$.

Example 1: Let $n = 1$. Consider the addition $S = U + V$ given that $U = 0$ and $V = 1$. We have 12 unknowns and $32n - 20 = 12$ equations in the rough system:

$$\begin{aligned} \mathrm{P}(1) = 0\ ; \quad \mathrm{P}(-2) = 0\ ; \\ \mathrm{P}(3) = \mathrm{P}(2; -1) + \mathrm{P}(-2; 1) = 0\ ; \quad \mathrm{P}(4) = \mathrm{P}(2; 1)\ ; \\ \mathrm{P}(-1) + \mathrm{P}(1) = 1\ ; \quad \mathrm{P}(-2) + \mathrm{P}(2) = 1\ ; \\ \mathrm{P}(-3) + \mathrm{P}(3) = 1\ ; \quad \mathrm{P}(-4) + \mathrm{P}(4) = 1\ ; \\ \mathrm{P}(2) = \mathrm{P}(2; -1) + \mathrm{P}(2; 1); \quad \mathrm{P}(-2) = \mathrm{P}(-2; -1) + \mathrm{P}(-2; 1)\ ; \\ \mathrm{P}(1) = \mathrm{P}(-2; 1) + \mathrm{P}(2; 1)\ ; \quad \mathrm{P}(-1) = \mathrm{P}(-2; -1) + \mathrm{P}(2; -1). \end{aligned} \tag{14}$$

The rank of the system is 11. When accounting for Eq. (12), two equations become identically zero and thus the four unknowns with more than one literal become irrelevant. We have then,

$$\begin{aligned} \mathrm{P}(1) = 0\ ;\ \mathrm{P}(-2) = 0\ ; \\ \mathrm{P}(3) = \mathrm{P}(-1)\ ;\ \mathrm{P}(4) = 0\ ; \\ \mathrm{P}(-1) = 1\ ;\ \mathrm{P}(2) = 1\ ;\ \mathrm{P}(-3) + \mathrm{P}(3) = 1\ ;\ \mathrm{P}(-4) + \mathrm{P}(4) = 1\ ;\ \mathrm{P}(-1) = \mathrm{P}(2). \end{aligned}$$

The first row is just the probability formulation of the data: $U = 0\,; V = 1$. The second row described the two structural equations. The last row displays the universal equations. The resolution is straightforward. The rank of the linear system is now $8 = 8n$ and the LP problem is feasible with the conventional deterministic solution $S = 1$.

Example 2: Let $n = 2$. Consider the addition $S = U + V$ given $U = 2$ and $V = 3$. We have $28n - 16 = 40$ unknowns with 44 equations, namely Eqs. (4, 9, 10, 11). The rank of the rough linear system is 35. The LP systeme is feasible. Using Eq.(12), the rank is at once lowered to $16 = 8n$ and all unknowns with more than one literal become irrelevant. After Gauss elimination the matrix is diagonal, so no LP algorithm is needed. This result seems general: The system is trivial when accounting for Eq.(12).

Example 3: Let $n = 1$. Consider the subtraction $V = S - U$ given that $S = 0$ and $U = 1$. The linear system Eq.(14) is still valid, except the first row which is now $\mathrm{P}(-1) = 0; \mathrm{P}(3) = 0; \mathrm{P}(4) = 0$. We have still 12 unknowns but $32n - 19 = 13$ equations. The rank of the linear system is 12 and the single solution is

$$\begin{array}{lllll} \mathrm{P}(-1) = 0\ ; & \mathrm{P}(-2) = 2\ ; & \mathrm{P}(-3) = 1\ ; & \mathrm{P}(-4) = 1\ ; \mathrm{P}(-2;1) = 1\ ; & \mathrm{P}(-2;-1) = 1\ ; \\ \mathrm{P}(1) \ = 1\ ; & \mathrm{P}(2) = -1\ ; & \mathrm{P}(3) = 0\ ; & \mathrm{P}(4) = 0\ ; \mathrm{P}(2;1) \ = 0\ ; & \mathrm{P}(2;-1) = -1. \end{array}$$

The LP systeme is not feasible because e.g. $\mathrm{P}(2) < 0$. When accounting for Eq.(12) the linear system is at once impossible. However, in general when $S < U$ the linear system is possible but the LP problem is not feasible.

Addition of several integers: The addition of several integers can be performed step by step. We start with an integer U_0. We add a first term U_1 to obtain a first sum S_1 and internal variables for the carry bits. Next, we add the second term U_2 to S_1, etc. We will use this process in the next section for the Bayesian multiplication.

Remark on the Bayesian addition. If we just have to add or subtract two integers, the Bayesian environment is certainly not a paragon of simplicity. For instance, we have seen that the sum of two integers of two bits is solved by a LP problem with 40 unknowns and 44 equations! The same conclusion can be drawn for any operation known to be of complexity P. Nevertheless, for exotic problems of the addition environment, if we have an input of n_b data bits and no algorithm except a *force brute* exploration of the 2^{4n-n_b} potential assignments, the method will be useful for large n. For what we are concerned, we will use these results in a more complex environment, namely, multiplication.

3 Multiplication

We now aim to encode the product of two integers A and B in the framework of Bayesian arithmetic. Most of the ingredients are directly derived from the previous section and thus we will just have to replicate the results with a minimum of details. Let n and m respectively be the number of bits of the two integers. When setting down a conventional binary multiplication we have simply to add m terms, U_0, U_1,..., U_{m-1}. The addition will be processed in $m - 1$ steps. In this context, the integers are 'shifted', that is U_0 is composed of n bits and the following terms U_t are also composed of n significant bits followed by t zeros.

3.1 Addition of shifted integers

Let $t \in [\![0, m-1]\!]$. We have

$$U_t = \sum_{i=0}^{n-1} u_{t,t+i} 2^{t+i}$$

Example Let us write the binary additions of shifted integers for $n = 2$ and $m = 3$. Let $r_{t,t+i}$ be the carry bits.

U_0				$\mathbf{u_{01}}$	$\mathbf{u_{00}}$
U_1			$\mathbf{u_{12}}$	$\mathbf{u_{11}}$	.
R_1		$\mathbf{r_{13}}$	$\mathbf{r_{12}}$	.	.
$S_1 = U_0 + U_1$		r_{13}	$\mathbf{s_{12}}$	$\mathbf{s_{11}}$	u_{00}
U_2		$\mathbf{u_{23}}$	$\mathbf{u_{22}}$	.	.
R_2	$\mathbf{r_{24}}$	$\mathbf{r_{23}}$	.	.	.
$S_2 = S_1 + U_2$	r_{24}	$\mathbf{s_{23}}$	$\mathbf{s_{22}}$	s_{11}	u_{00}

We note that we have n significant bits for each integer U_0, U_t, S_t and a 'carry integer' R_t for $t \in [\![1, m-1]\!]$ (displayed in boldface). Let $\mathsf{U}_{0,i}$, $\mathsf{U}_{t,t+i}$, $\mathsf{S}_{t,t+i}$ and $\mathsf{R}_{t,t+i+1}$ be the random variables describing these integers for $i \in [\![0, n-1]\!]$. Again, it is suitable to bring together these $3nm - 2n$ variables in a single list X_k of *global variables*, labelled from $k = 1$ to $3nm - 2n$. We will adopt the labelling convention of table 4.

Example When $n = 2$ and $m = 3$ the shifted addition reads,

$\mathsf{U}_{0,i}$	$\mathsf{U}_{t,i}$	$\mathsf{S}_{t,i}$	$\mathsf{R}_{t,i}$
$i \in [\![0, n-1]\!]$	$i \in [\![t, t+n-1]\!]$	$i \in [\![t, t+n-1]\!]$	$i \in [\![t+1, t+n]\!]$
$k = i+1$	$k = n(3t-2)+i-t+1$	$k = n(3t-1)+i-t+1$	$k = 3nt+i-t$
$k \in [\![1, n]\!]$	$k \in [\![(3t-2)n+1, (3t-1)n]\!]$	$k \in [\![(3t-1)n+1, 3tn]\!]$	$k \in [\![3tn+1, n(3t+1)]\!]$
X_1 to X_n	$\mathsf{X}_{(3t-2)n+1}$ to $\mathsf{X}_{(3t-1)n}$	$\mathsf{X}_{(3t-1)n+1}$ to X_{3tn}	X_{3tn+1} to $\mathsf{X}_{n(3t+1)}$

Table 4: Labelling convention of the global variables X_k corresponding to the addition of m shifted integers of n bits. We have $t \in [\![1, m-1]\!]$ and a total of $3nm - 2n$ variables.

$$
\begin{aligned}
\mathrm{P}(2n+1) &= \mathrm{P}(-2; n+1) + \mathrm{P}(2; -n-1) \\
\mathrm{P}(3n+1) &= \mathrm{P}(2; n+1) \\
\mathrm{P}(2n+i) &= \mathrm{P}(i+1; -i-n; -i-3n+1) + \mathrm{P}(-i-1; i+n; -i-3n+1) + \mathrm{P}(-i-1; -i-n; i+3n-1) + \mathrm{P}(i+1; i+n; i+3n-1) \\
\mathrm{P}(3n+i) &= \mathrm{P}(i+1; i+n; -i-3n+1) + \mathrm{P}(-i-1; i+n; i+3n-1) + \mathrm{P}(i+1; -i-n; i+3n-1) + \mathrm{P}(i+1; i+n; i+3n-1) \\
\mathrm{P}(3n) &= \mathrm{P}(2n; -4n+1) + \mathrm{P}(-2n; 4n-1) \\
\mathrm{P}(4n) &= \mathrm{P}(2n; 4n-1) \\
\mathrm{P}(3nt-n+1) &= \mathrm{P}(3nt-4n+2; -3nt+2n-1) + \mathrm{P}(-3nt+4n-2; 3nt-2n+1) \\
\mathrm{P}(3nt+1) &= \mathrm{P}(3nt-4n+2; 3nt-2n+1). \\
\mathrm{P}(3nt-n+i) &= \mathrm{P}(-3nt+4n-i-1; -3nt+2n-i; 3nt+i-1) + \mathrm{P}(-3nt+4n-i-1; 3nt-2n+i; -3nt-i+1) \\
&\quad + \mathrm{P}(3nt-4n+i+1; -3nt+2n-i; -3nt-i+1) + \mathrm{P}(3nt-4n+i+1; 3nt-2n+i; 3nt+i-1) \\
\mathrm{P}(3nt+i) &= \mathrm{P}(3nt-4n+i+1; 3nt-2n+i; -3nt-i+1) + \mathrm{P}(3nt-4n+i+1; -3nt+2n-i; 3nt+i-1) \\
&\quad + \mathrm{P}(-3nt+4n-i-1; 3nt-2n+i; 3nt+i-1) + \mathrm{P}(3nt-4n+i+1; 3nt-2n+i; 3nt+i-1) \\
\mathrm{P}(3nt) &= \mathrm{P}(-3nt+2n; -3nt+n; 3nt+n-1) + \mathrm{P}(-3nt+2n; 3nt-n; -3nt-n+1) \\
&\quad + \mathrm{P}(3nt-2n; -3nt+n; -3nt-n+1) + \mathrm{P}(3nt-2n; 3nt-n; 3nt+n-1) \\
\mathrm{P}(3nt+n) &= \mathrm{P}(3nt-2n; 3nt-n; -3nt-n+1) + \mathrm{P}(3nt-2n; -3nt+n; 3nt+n-1) \\
&\quad + \mathrm{P}(-3nt+2n; 3nt-n; 3nt+n-1) + \mathrm{P}(3nt-2n; 3nt-n; 3nt+n-1)
\end{aligned}
$$

Table 5: Structural equations of the addition of m shifted integers of n bits. ($i \in [\![2, n-1]\!]$; $t \in [\![2, m-1]\!]$). We have a total of $2n(m-1)$ equations.

U_0				$\mathsf{X_2}$	$\mathsf{X_1}$
U_1			$\mathsf{X_4}$	$\mathsf{X_3}$	.
R_1		$\mathsf{X_8}$	$\mathsf{X_7}$	.	.
$S_1 = U_0 + U_1$		$\mathsf{X_8}$	$\mathsf{X_6}$	$\mathsf{X_5}$	$\mathsf{X_1}$
U_2		$\mathsf{X_{10}}$	$\mathsf{X_9}$	.	.
R_2	$\mathsf{X_{14}}$	$\mathsf{X_{13}}$	.	.	.
$S_2 = S_1 + U_2$	$\mathsf{X_{14}}$	$\mathsf{X_{12}}$	$\mathsf{X_{11}}$	$\mathsf{X_5}$	$\mathsf{X_1}$

Structural equations Since we proceed with the computation step by step, we just have to bring together the structural equations of each step. We extract from the full operation the relevant computation section and apply the previous result concerning the addition of two integers.

First step: For $t = 1$, the relevant computation section is the following,

U_0			$\mathbf{u_{0,n-1}}$	$\cdots$	$\mathbf{u_{0,i}}$	$\cdots$	$\mathbf{u_{0,2}}$	$\mathbf{u_{0,1}}$
U_1		$\mathbf{u_{1,n}}$	$\mathbf{u_{1,n-1}}$	$\cdots$	$\mathbf{u_{1,i}}$	$\cdots$	$\mathbf{u_{1,2}}$	$\mathbf{u_{1,1}}$
R_1	$\mathbf{r_{1,n+1}}$	$\mathbf{r_{1,n}}$	$\mathbf{r_{1,n-1}}$	$\cdots$	$\mathbf{r_{1,i}}$	$\cdots$	$\mathbf{r_{1,2}}$	.
S_1	$r_{1,n+1}$	$\mathbf{s_{1,n}}$	$\mathbf{s_{1,n-1}}$	$\cdots$	$\mathbf{s_{1,i}}$	$\cdots$	$\mathbf{s_{1,2}}$	$\mathbf{s_{1,1}}$

Next steps: For $t \in [\![2, m-1]\!]$, the computation section reads,

S_{t-1}		$r_{t-1,t+n-1}$	$\cdots$	$\mathbf{s_{t-1,t+i}}$	$\cdots$	$\mathbf{s_{t-1,t+1}}$	$\mathbf{s_{t-1,t}}$
U_t		$\mathbf{u_{t,t+n-1}}$	$\cdots$	$\mathbf{u_{t,t+i}}$	$\cdots$	$\mathbf{u_{t,t+1}}$	$\mathbf{u_{t,t}}$
R_t	$\mathbf{r_{t,t+n}}$	$\mathbf{r_{t,t+n-1}}$	$\cdots$	$\mathbf{r_{t,t+i}}$	$\cdots$	$\mathbf{r_{t,t+1}}$	.
S_t	$r_{t,t+n}$	$\mathbf{s_{t,t+n-1}}$	$\cdots$	$\mathbf{s_{t,t+i}}$	$\cdots$	$\mathbf{s_{t,t+1}}$	$\mathbf{s_{t,t}}$

Collecting the results of each step, be obtain the set of structural equations expressed in terms of global variables given in table 5. We have $2n$ equations per section and $m-1$ sections, i.e., a total of $2n(m-1)$ structural equations.

$\mathrm{P}(k)$
$\mathrm{P}(2; n+1); \mathrm{P}(2n; 4n-1)$; $\mathrm{P}(3nt-4n+2; 3nt-2n+1)$
$\mathrm{P}(i+n; i+3n-1)$; $\mathrm{P}(i+1; i+3n-1)$; $\mathrm{P}(i+1; i+n)$
$\mathrm{P}(3nt-2n+i; 3nt+i-1)$; $\mathrm{P}(3nt-4n+i+1; 3nt+i-1)$; $\mathrm{P}(3nt-4n+i+1; 3nt-2n+i)$
$\mathrm{P}(3nt-n; 3nt+n-1)$; $\mathrm{P}(3nt-2n; 3nt+n-1)$; $\mathrm{P}(3nt-2n; 3nt-n)$
$\mathrm{P}(i+1; i+n; i+3n-1)$; $\mathrm{P}(3nt-2n; 3nt-n; 3nt+n-1)$
$\mathrm{P}(3nt-4n+i+1; 3nt-2n+i; 3nt+i-1)$

Table 6: Positive unknowns of the addition of m shifted integers of n bits. ($i \in [\![2, n-1]\!]$; $t \in [\![2, m-1]\!]$; $k \in [\![1, n(3m-2)]\!]$). We have a total of $7mn - 3m - 6n$ positive unknowns.

Example: For $n = 2$ and $m = 3$, we have $2n(m-1) = 8$ structural equations:

$$\begin{aligned}
\mathrm{P}(5) &= \mathrm{P}(-2; 3) + \mathrm{P}(2; -3) \\
\mathrm{P}(7) &= \mathrm{P}(2; 3) \\
\mathrm{P}(6) &= \mathrm{P}(4; -7) + \mathrm{P}(-4; 7) \\
\mathrm{P}(8) &= \mathrm{P}(4; 7) \\
\mathrm{P}(11) &= \mathrm{P}(6; -9) + \mathrm{P}(-6; 9) \\
\mathrm{P}(13) &= \mathrm{P}(6; 9) \\
\mathrm{P}(12) &= \mathrm{P}(-8; -10; 13) + \mathrm{P}(-8; 10; -13) + \mathrm{P}(8; -10; -13) + \mathrm{P}(8; 10; 13) \\
\mathrm{P}(14) &= \mathrm{P}(8; 10; -13) + \mathrm{P}(8; -10; 13) + \mathrm{P}(-8; 10; 13) + \mathrm{P}(8; 10; 13)
\end{aligned}$$

Relevant unknowns In order to list the relevant unknowns, we inspect the structural equations. We have $(mn - m - n)$ 3-literal positive unknowns and then $3 \times (mn - m - n)$ 2-literal positive variants. We have m direct 2-literal positive unknowns and finally $(3mn - 2n)$ 1-literal positive unknowns. Gathering together these results we have a total $7mn - 3m - 6n$ positive unknowns and a total $26mn - 16m - 24n$ unknowns (Table 6).

literals	positive unknowns	unknowns
1	$3mn - 2n$	$6mn - 4n$
2	$3mn - 2m - 3n$	$12mn - 8m - 12n$
3	$mn - m - n$	$8mn - 8m - 8n$
Total	$7mn - 3m - 6n$	$26mn - 16m - 24n$

Example: Let $n = 2$ and $m = 3$, we have $7mn - 3m - 6n = 21$ relevant positive unknowns,

$\mathrm{P}(k), k = 1$ to 14

$\mathrm{P}(2; 3)$, $\mathrm{P}(7; 4)$, $\mathrm{P}(6; 9)$, $\mathrm{P}(10; 13)$, $\mathrm{P}(8; 13)$, $\mathrm{P}(8; 10)$

$\mathrm{P}(8; 10; 13)$

Relevant universal equations The relevant universal equations are derived from the relevant positive unknowns by Eq.(6, 7, 8).

literals	positive unknowns	universal equations
1	$3mn - 2n$	$3mn - 2n$
2	$3mn - 2m - 3n$	$12mn - 8m - 12n$
3	$mn - m - n$	$12mn - 12m - 12n$
Total	$7mn - 3m - 6n$	$27mn - 20m - 24n$

For instance, for $n = 2$ et $m = 3$, we have $27mn - 20m - 24n = 54$ relevant universal equations.

We have completed the analysis of the addition of m shifted integers of n bits. We have identified $26mn - 16m - 24n$ relevant unknowns, $2n(m-1)$ structural equations and $27mn - 20m - 24n$ universal equations, but not defined any data equation. We are now ready to apply these results to the very multiplication.

3.2 Multiplication

Let A and B be two integers with binary expansions,

$$A = \sum_{i=0}^{n-1} 2^i a_i \quad ; \quad B = \sum_{t=0}^{m-1} 2^t b_t,$$

A_0 to A_{n-1}	B_0 to B_{m-1}
$i \in [\![0, n-1]\!]$	$t \in [\![0, m-1]\!]$
$k = i + 3nm - 2n + 1$	$k = t + 3nm - n + 1$
$k \in [\![3nm - 2n + 1, 3nm - n]\!]$	$k \in [\![3nm - n + 1, 3nm - n + m]\!]$
$\mathsf{X}_{3nm-2n+1}$ to X_{3nm-n}	$\mathsf{X}_{3nm-n+1}$ to $\mathsf{X}_{3nm-n+m}$

Table 7: Labelling convention of the global variables X_k corresponding to the variables A_i, B_t. This defines a number of mn variables.

C_0	C_1 to C_{m-1}	C_m to C_{m+n-2}	C_{m+n-1}
$j = 0$	$j \in [\![1, m-1]\!]$	$j \in [\![m, m+n-2]\!]$	$j = m + n - 1$
$k = 1$	$k = 3nj - n + 1$	$k = 3nm - 4n - m + 2 + j$	$k = 3nm - 2n$
X_1	X_{2n+1} to $\mathsf{X}_{3nm-4n+1}$	$\mathsf{X}_{3nm-4n+2}$ to X_{3nm-3n}	X_{3nm-2n}

Table 8: Labelling convention of the global variables X_k corresponding to the variables C_j, already defined in Table 4 to describe the variables $\mathsf{U}_{0,0}$, $\mathsf{S}_{t,t}$, $\mathsf{S}_{m-1,j}$ and $\mathsf{R}_{m-1,m+n-1}$.

where n and m are the number of bits of A and B respectively. Let C be a third integer and Π the logical proposition

$$\Pi : \quad A \times B = C.$$

We suppose throughout that Π is satisfied. The binary expansion of C reads

$$C = \sum_{i=0}^{n-1} \sum_{t=0}^{m-1} 2^{i+t} a_i b_t = \sum_{j=0}^{n+m-1} 2^j c_j$$

For $t \in [\![0, m-1]\!]$, define U_t,

$$U_t = \sum_{i=0}^{n-1} 2^{i+t} a_i b_t = \sum_{i=t}^{t+n-1} u_{t,i} 2^i \quad \text{so that} \quad C = \sum_{t=0}^{m-1} U_t$$

Clearly, the integers U_t form a set of m shifted integers as described in the previous section. Thus, we will take back the random variables $\mathsf{U}_{t,i}$, $\mathsf{S}_{t,i}$, $\mathsf{R}_{t,i}$. We will also define new random variables A_i and B_t corresponding to the binary expansion of A and B. The binary variables of C are already ranked because $C = S_{m-1}$. Eventually, we construct a probability space with all these variables, namely $\mathsf{U}_{t,i}$, $\mathsf{S}_{t,i}$, $\mathsf{R}_{t,i}$, A_i and B_t, and define a Bayesian probability distribution P given Π.

It is necessary to extend the list of global variables X_k, (Table 4), in order to account for A and B. We will adopt the convention of Table 7. The global variables corresponding to the bits of $C = S_{m-1}$ are recalled in Table 8.

Example Let $n = 2$ and $m = 3$. We have

$\mathsf{A}_0 = \mathsf{X}_{15}$; $\mathsf{A}_1 = \mathsf{X}_{16}$.

$\mathsf{B}_0 = \mathsf{X}_{17}$; $\mathsf{B}_1 = \mathsf{X}_{18}$; $\mathsf{B}_2 = \mathsf{X}_{19}$;

$\mathsf{C}_0 = \mathsf{X}_1$; $\mathsf{C}_1 = \mathsf{X}_5$; $\mathsf{C}_2 = \mathsf{X}_{11}$; $\mathsf{C}_3 = \mathsf{X}_{12}$; $\mathsf{C}_4 = \mathsf{X}_{14}$;

Structural equations In addition to the structural equations of Table 5, let us consider the truth table of the variables $\mathsf{U}_{t,t+i}$ versus A_i and B_t.

B_t	A_i	$\mathsf{U}_{t,t+i}$
0	0	0
0	1	0
1	0	0
1	1	1

The codification of this truth table into linear equations is straightforward.

$$\mathrm{P}(\mathsf{U}_{t,t+i} = 1 | \Pi) = \mathrm{P}(\mathsf{A}_i = 1; \mathsf{B}_t = 1 | \Pi)$$

Thus, we obtain a set of mn new structural equations displayed with the global variable convention in Table 9, to be added to the $2mn - 2n$ equations of table 5 and leading to a total of $3mn - 2n$ structural equations.

$$\mathrm{P}(i) = \mathrm{P}(i + 3nm - 2n + 1; 3nm - n + m + 3)$$
$$\mathrm{P}(n(3t-2) + i - t + 1) = \mathrm{P}(i + 3nm - 2n + 1; t + 3nm - n + m + 3)$$

Table 9: Complementary structural equations of the multiplication ($i \in [\![0, n-1]\!]$, $t \in [\![1, m-1]\!]$) in addition to Table 5. The number of new equations is mn and the total number of structural equations is $3nm - 2n$.

$$\mathrm{P}(k)$$
$$\mathrm{P}(3nm - 2n + 1 + i; 3nm - n + 1 + t)$$

Table 10: Complementary positive unknowns of the multiplication in addition to Table 6. ($i \in [\![0, n-1]\!]$; $t \in [\![0, m-1]\!]$; $k \in [\![3nm - 2n + 1, 3nm - n + m]\!]$). We have $mn + m + n$ complementary positive unknowns and a total of $8mn - 2m - 5n$ positive unknowns.

Relevant unknowns Again, we take back the relevant unknowns of the shifted addition (Table 6). We have to add $n + m$ new positive relevant unknowns of one literal, namely $\mathrm{P}(\mathsf{A}_i|\Pi)$ and $\mathrm{P}(\mathsf{B}_t|\Pi)$ and nm new positive relevant unknowns of two literals, namely $\mathrm{P}(\mathsf{A}_i; \mathsf{B}_t|\Pi)$. They are listed in Table 10.

Example: Let $n = 2$ and $m = 3$. The 11 new relevant positive unknowns are,

P(15) ; P(16) ; P(17); P(18) ; P(19)

P(19;15) ; P(19;16) ; P(18;15) ; P(18;16) ; P(17;15) ; P(17;16)

The unknowns are derived from the positive unknowns from enumeration of the variants.

literal	positive unknowns	unknowns	universal equations
1	$3mn + m - n$	$6mn + 2m - 2n$	$3mn + m - n$
2	$4mn - 2m - 3n$	$16mn - 8m - 12n$	$16mn - 8m - 12n$
3	$mn - m - n$	$8mn - 8m - 8n$	$12mn - 12m - 12n$
Total	$8mn - 2m - 5n$	$30mn - 14m - 22n$	$31mn - 19m - 25$

Relevant universal equations We derive the relevant universal equations from the list of positive unknowns. We have a total of $1\times(3nm-n+m)+4\times(4mn-2m-3n)+12\times(mn-m-n) = 31mn-19m-25n$ relevant universal equations.

In summary, for $m, n > 1$ we have

unknowns:	$30mn - 14m - 22n$
structural equations:	$3mn - 2n$
universal equations:	$31mn - 19m - 25n$

3.3 Polynomial time algorithm of factorization

We are now able to factorize an integer in polynomial time. Let $C > 3$ be a given integer of c bits, that is $2^{c-1} \leq C < 2^c$. Let A and B be two unknown factors so that $C = A \times B$. Let a and b be the number of bits of A and B respectively. We have $a + b - 1 \leq c \leq a + b$. If $B \leq A$, the trivial solution is $A = C$ and $B = 1$, i.e., $a = c$ and $b = 1$. All non trivial solutions with $B \leq A$ are such that $a < c$ and $b \leq c/2$. In the multiplication environment, we can choose $n = c - 1 > 1$ and $m = [(c + 1)/2] > 1$, given that we may complete the sets of bits by a number of zeros if necessary. Since we have $m + n > c$, it is convenient to define the array $c_j = 0$ for $j \in [\![c, m + n - 1]\!]$. Now, we construct the LP system of the multiplication environment. The system of both structural (Tables 5, 9) and universal equations is complemented with the $m + n$ following *specific data equations*:

$$\mathrm{P}(\mathsf{C}_j = c_j|\Pi) = 1$$

where $j \in [\![0, m + n - 1]\!]$. These data equations are translated in Table 11 in term of global variables X_k (using Table 4).

We have a LP problem of $30mn - 14m - 22$ unknowns and $34mn - 18m - 26n$ equations. Note that the equations have a maximum of 3 coefficients with an average of 2 entries. Therefore, the matrix is widely sparse. Since $m = \mathrm{O}(c)$ and $n = \mathrm{O}(c)$ the dimension of the problem is $\mathrm{O}(c^2)$. The detail is the following:

$$
\begin{aligned}
\mathrm{P}(1) &= c_0 \\
\mathrm{P}(3nj - n + 1) &= c_j \text{ for } j \in [\![1, m-1]\!] \\
\mathrm{P}(3nm - 4n - m + 2 + j) &= c_j \text{ for } j \in [\![m, m+n-2]\!] \\
\mathrm{P}(3nm - 2n) &= c_{m+n-1}
\end{aligned}
$$

Table 11: Data specific equations for factorization. The numbers c_j are the coefficients of the binary expansion of the input integer C for $j < c$ and 0 for $j \geq c$. These $n + m$ equations have to be added to the set of structural equations (Tables 5, 9) and to the universal equations.

unknowns:	$30mn - 14m - 22n$
including	
1 literal:	$2 \times (3mn + m - n)$
2 literals:	$4 \times (4mn - 2m - 3n)$
3 literals:	$8 \times (mn - m - n)$
equations:	$34mn - 18m - 26n$
including	
structural:	$3mn - 2n$
universal:	$31mn - 19m - 25n$
data:	$m + n$

It is suitable to simplify at once the system by use of Eq.(12).

- If the system is feasible, we have to check whether deterministic solutions exist by optimization. Generally, this can be obtained by optimazing a maximum of $2m$ objective functions. Let $k_0, k_1, \ldots, k_{m-1}$ be the global labels corresponding to the variables $\mathsf{B}_0, \mathsf{B}_1 \ldots \mathsf{B}_{m-1}$ respectively (Table 7). We first select the two objective functions $z_0 = \mathrm{P}(\pm k_0)$. If the two maxima are both different from 1, then C is prime. Otherwise, we have a maximum $z_1 = 1$ for say $z_0 = \mathrm{P}(-k_0)$. Now, we select two objective functions $z_1 = \mathrm{P}(-k_0) + \mathrm{P}(\pm k_1)$. Now, if the two maxima are both different from 2, then again C is prime. Otherwise the maximum 2 is obtained e.g., for $z_1 = \mathrm{P}(-k_0) + \mathrm{P}(k_1)$. We iterate with $z_2 = \mathrm{P}(-k_0) + \mathrm{P}(k_1) + \mathrm{P}(\pm k_2)$, etc. Finally, we select an integer B with $\mathrm{P}(\pm k_i) = 1$ for $i \in [\![0, m-1]\!]$ or otherwise C is prime. Finally, we check C/B. Thus, we obtain two factors A and B or prove that C is prime in polynomial time.

- If the system is not feasible, C is prime with certainty. Again, this result is obtained in polynomial time. (However, from our computations, the system seems always feasible).

Example 1: A trivial problem. Let $C = 6$ and thus $c = 3$ bits. We can choose $m = 2$ and $n = 2$. Even if the computation is straightforward, we have nevertheless 48 unknowns and 48 equations and therefore the computation is difficult to perform by hand! After Gauss elimination, the rank is 42. The LP problem is feasible and since $m = n$ the factors A and B are not implicitly ordered. We obtain the expected conventional deterministic solutions ($A = 2$; $B = 3$) and ($A = 3$; $B = 2$) but also a continuous set of non deterministic solutions which can be regarded as a superposition of the deterministic solutions (in the quantum sense).

Example 2: A toy model for quantum mechanics. Let $C = 5$ and thus $c = 3$ bits. We can choose $m = 2$ and $n = 2$. We have still 48 unknowns, 48 equations and the rank is 42. Again, the LP problem is feasible but only accept *non-fully deterministic solution*. For instance, if we have $\mathrm{P}(\mathsf{A}_0 = 1|\Pi) = 1$ and $\mathrm{P}(\mathsf{B}_0 = 1|\Pi) = 1$, we do have $\mathrm{P}(\mathsf{A}_1 = 1|\Pi) = 1/2$ and $\mathrm{P}(\mathsf{B}_1 = 1|\Pi) = 1/2$. Let us open a parenthesis: When contemplating the LP problem, it resembles a quantum system. We give a few hints even if a comprehensive discussion is clearly beyond the scope of this paper: The prime '5' is a mathematical object defined by the variables C_j. We decide to describe this object by a number of artificial variables, like A_i and B_i. As a result, the outcomes of these new variables may be only defined in probability, depending upon the solution of the LP system. Each solution is similar to the setting of a quantum object. When a particular solution/setting is chosen, the probabilities are *ipso facto* determined. They rely on the structure of internal parameters, described, e.g., by the symmetry of the feasible LP-polytope [7]. In order to force a deterministic outcome, we may proceed to a random trial but then, the system collapses, e.g., into two integers and the original prime is destroyed. In this respect the LP system of a composite integer like '6' resembles a classical object with some quantum features like possible superposition of states. In our opinion, this analogy supports the conjecture that quantum formalism is a particular codification of states of knowledge, exactly as the Bayesian formulation is. We close this parenthesis.

For small values of C, when compared with conventional algorithms, one can quite rightly argue that the computation complexity is out of all proportions, up to say 100 bits. Actually, the interest of the method only arises for big integers, when the conventional algorithms crash into the exponential wall, while the Bayesian route remains polynomial. The next examples are potentially in this range but are not in the present capability of this author due at least to lack of familiarity with large systems.

Example 3: The present state of the art. Let C be a 768-bit integer. We can choose $m = 384$ and $n = 767$. We obtain a rough LP system with 8 813 590 unknowns and 9 987 098 equations which uses sparse matrices with three or less non-zero entries per row. The number of 768 bits corresponds to the last factorized integer of the now obsolete *RSA Challenge* [9]. The computation was carried out in 2009 by an international team [10], using the best known algorithm, namely, the *Number Field Sieve* (NFS) [11].

According to the authors, the overall effort represent 2000 years on a single core 2.2 GHz processor. Most of the computation consists in sieving a large number of smooth roots modulo C from a pair of convenient polynomials. The last operation is the merging step for Gauss elimination. It produced a 192 796 550 $\times$ 192 795 550 sparse matrix with on average 144 non-zero entries per row solved in one day. Clearly, the present Bayesian method is in principle by far simplest than this last operation.

Example 4: A challenging computation beyond the present state of the art. Let C be a 1024-bit integer. We can choose $m = 512$ and $n = 1023$. We obtain a rough LP sparse system with 15 683 606 unknowns and 17 772 570 equations still with three or less non-zero entries per row. These dimensions may be reduced if we know the order of magnitude of the factors. The computation is only twice more difficult than for 768 bits. Using the NFS algorithm, the factorization of the 1024-bit integer of the RSA challenge is presently out of reach but expected by the year 2020 .

Exemple 5: An outstanding challenging computation. Let C be a 2048-bit integer. The rough system dimensions will be about $63.\ 10^6 \times 71.\ 10^6$. This factorization is definitively out of reach of the NFS algorithm but, in principle, still tractable with the present Bayesian method.

4 Conclusion

The paper applies previous results on the conversion of an ensemble of Boolean formulae into a linear programming problem: Now, we introduce the concept of 'Bayesian arithmetic'. In this model, a Diophantine equation is interpreted as a set of prior conditions in the framework of Bayesian probability theory. We have shown that the dimension of the LP problem scales as $\mathrm{O}(n)$ for the sum of two n-bit integers and as $\mathrm{O}(mn)$ for the product of two integers of m and n bits respectively. As a result, the LP problem encoding a Diophantine equation is solved in polynomial time in the number of working bits.

The first significant application is a polynomial time algorithm of factorization. This stresses the need to revisit all current public key encryption systems.

A similar approach should solve in polynomial time any logical function defined by a set of truth tables.

Finally, the exponential speed-up over conventional algorithms recalls quantum computation. This questions about a deep similarity between LP and quantum mechanics formalism.

References

[1] M. Feldmann, From classical versus quantum algorithms to P versus NP (May 2012). `arXiv:1205.6658[cs.CC]`.

[2] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J.Sci.Statist.Comput. 26 (1997) 1484. `arXiv:quant-ph/9508027`.

[3] M. Agrawal, N. Kayal, N. Saxena, PRIME is in P, Annals of Mathematics 160 (2004) 781–793.
URL `http://annals.math.princeton.edu/wp-content/uploads/annals-v160-n2-p12.pdf`

[4] E. T. Jaynes, Probability Theory: The Logic of Science, Cambridge University Press, Cambridge, UK, 2003.

[5] R. T. Cox, Probability, frequency, and reasonable expectation, American Journal of Physics 14 (1946) 1–13.

[6] T. H. Cormen, C. E. Leiserton, R. L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, Cambridge,Ma., USA, 2009.

[7] K. G. Murty, Linear Programming, John Wiley & Sons, New York, 1983.

[8] L. Khachiyan, A polynomial time algorithm for linear programming, Doklady Akod. Nouk SSSR 244 (5) (1979) 1093–1096.

[9] RSA Laboratories, The RSA challenge (2012).
URL `http://www.rsa.com/rsalabs/node.asp?id=2093`

[10] T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. te Riele, A. Timofeev, P. Zimmermann, Factorization of a 768-bit RSA modulus (2010).
URL `http://eprint.iacr.org/2010/006.pdf`

[11] J. Lenstra, H. W., The development of the number field sieve, Lecture Notes in Math. 1–3 (1993) 1554.