Final results on P vs NP via Integer Factorization and Optimization

Yuly Shipilevsky

Toronto, Ontario, Canada
E-mail address: yulysh2000@yahoo.ca

Abstract

We reduce integer factorization problem to the NP-hard problem of minimizing a quadratic polynomial with integer coefficients over the integer points in a quadratically constrained two-dimensional region.

Next, we reduce integer factorization problem to the problem of enumeration of vertices of integer hull of a special two-dimensional rational polyhedron, solvable in time polynomial by Hartmann's algorithm.

Finally, as we find a polynomial-time algorithm to solve an NP-hard problem, we conclude that P = NP.

Keywords: integer factorization, integer programming, polynomial-time, NP-hard, rational polyhedron, integer hull

1 Introduction

Cryptography, elliptic curves, algebraic number theory have been brought to bear on integer factorization problem.

Until now, no algorithm has been published that can factor in deterministic polynomial time. For an ordinary computer the best published asymptotic running time is for the general number field sieve (GNFS) algorithm(see, e.g., A. K. Lenstra and H. W. Jr. Lenstra [10], P. Stevenhagen [12]).

The purpose of this paper is to develop a polynomial-time integer factorization algorithm, factoring in deterministic polynomial time, and, then, make more general conclusion: P = NP(see, e.g., Cormen et al. [3]).

The plan of this paper is as follows. In Section 2 we reduce integer factorization problem to some two-dimensional integer minimization problem and show that if there exists a nontrivial divisor of N, those divisor is a minimizer of those two-dimensional integer minimization problem, and any minimizer of those integer minimization problem is a nontrivial divisor of N.

We show that those two-dimensional integer minimization problem is NP-hard problem.

In Section 3 we construct a special two-dimensional rational polyhedron and reduce those NP-hard integer minimization problem to the integer minimization problem over the integer points in that rational polyhedron and show that despite generally, such problems can be solved in time polynomial, in our case it can't.

We develop a polynomial-time algorithm for integer factorization by enumeration of vertices of integer hull of that two-dimensional rational polyhedron.

We conclude that since we found a polynomial-time algorithm to solve an NP-hard problem, it would mean that P is equal NP.

2 Reduction to the Integer Programming problem. Minimum Principle

Let us reduce integer factorization problem to some integer minimization problem, so that any minimizer that is found solves integer factorization problem.

The key idea is to construct the objective function and constraints so that any minimizer satisfies the equation: $xy = N$, and, therefore, is a solution of the integer factorization problem.

Let us consider the following integer minimization problem:

$$\begin{aligned} &\text{minimize} && xy \\ &\text{subject to} && xy \geq N, \qquad\qquad (1) \\ & && 2 \leq x \leq N-1, \\ & && N/(N-1) \leq y \leq N/2, \\ & && x \in \mathbf{N},\ y \in \mathbf{N},\ N \in \mathbf{N}. \end{aligned}$$

Let $\Omega := \{ (x, y) \in \mathbf{R}^2 \mid xy \geq N,\ 2 \leq x \leq N-1, N/(N-1) \leq y \leq N/2,$ $x \in \mathbf{R},\ y \in \mathbf{R} \}$ for a given $N \in \mathbf{N}$.

Hence, $\Omega^I := \Omega \cap \mathbf{Z}^2$ is a feasible set of the problem (1).

It is clear that if there exists a nontrivial solution of integer factorization problem $xy = N$, the objective function: $f(x, y) = xy$ reaches minimum at the integer point of the border $xy = N$ of the region Ω and if there exists a nontrivial solution of integer factorization problem, any minimizer of the problem (1) provides a (nontrivial) solution of integer factorization problem.

Thus, in this case, any minimizer of the problem (1) guarantees solution of integer factorization problem and there exists at least one such minimizer.

Theorem 1(Minimum Principle). *If there exists a nontrivial solution of integer factorization prbblem, that solution is a minimizer of problem (1) and if there exists a nontrivial solution of integer factorization problem, any minimizer of the problem (1) is a nontrivial solution of integer factorization problem.*

As a result, we obtain the following Integer Factorization Algorithm.

Algorithm 1(Integer Factorization Algorithm).
Input: A positive integer number N.
Output: A nontrivial divisor of N(if it exists).

Solve the problem (1):
Based on the input data compute a minimizer (x_{min}, y_{min}) of the problem (1).
if ($x_{min}\, y_{min} = N$)
then
    Return a nontrivial divisor x_{min} **of N**
else
    Return "N is a prime"

Let us determine the complexity of the problem (1).

Despite in general, integer programming is NP-hard or even incomputable (see, e.g., Hemmecke et al. [7]), for some subclasses of target functions and constraints it can be computed in time polynomial.

Note that the dimension of the problem (1) is fixed and is equal to 2.

A fixed-dimensional polynomial minimization in integer variables, where the objective function is a convex polynomial and the convex feasible set is described by arbitrary polynomials can be solved in time polynomial(see, e.g ., Khachiyan and Porkolab [8]).

A fixed-dimensional polynomial minimization over the integer variables, where the objective function $f_0(x)$ is a quasiconvex polynomial with integer coefficients and where the constraints are inequalities $f_i(x) \le 0$, $i = 1, \dots, k$ with quasiconvex polynomials $f_i(x)$ with integer coefficients, $f_i : \mathbf{R}^n \to \mathbf{R}$, $f_i(x)$, $i = 0, \dots, k$ are polynomials of degree at most $p \ge 2$, can be solved in time polynomial in the degrees and the binary encoding of the coefficients(s-ee, e.g., Heinz [6], Hemmecke et al. [7], Lee [9]). Note that the degrees are unary encoded here as well as the number of the constraints.

A mixed-integer minimization of a convex function in a convex, bounded feasible set can be done in time polynomial, according to Baes et al. [2], Oe-rtel et al. [11].

Since the objective function $f(x, y) = xy$ of the problem (1) is a quasico-ncave function in the feasible set Ω of the problem (1), we cannot use the re-sults described in Baes et al. [2], Heinz [6], Hemmecke et al. [7], Khachiyan and Porkolab [8], Oertel et al. [11] in order to solve the problem (1) in time polynomial in log(N). Note that Ω^I is described by quasiconvex polynomi-als, since $(-xy + N)$ is a quasiconvex function for $x > 0$, $y > 0$.

In general, since variables $x \in \mathbf{N}$, $y \in \mathbf{N}$ are bounded by the finite bounds $2 \le x \le N - 1$, $N/(N-1) \le y \le N/2$, the problem (1) and the respective Algorithm 1 are computable(see, e.g., Hemmecke et al. [7]), but still are NP-hard, since the problem (1) is a quadratically constrained integer minimizati-on problem(see, e.g., Del Pia and Weismantel [4], Del Pia et al. [5]).

Note that NP-hardness of (1) is clearly confirmed, e.g. in Del Pia et al. [5] : "... Using the same reduction as Lemma 1.2, it is possible to show that pro-blem (1) is NP-hard even when n = d = 2, P is a bounded, rational polyhedr-on, and we add a single quadratic inequality constraint (see [18]) ...".

3 Linearization. Polynomial-time Integer Factorization. P = NP

It was shown in Del Pia and Weismantel [4] that problem of minimizing a quadratic polynomial with integer coefficients over the integer points in a g-eneral two-dimensional rational polyhedron is solvable in time bounded by a polynomial in the input size and it was further extended to cubic and homog-eneous polynomials in Del Pia et al. [5].

Del Pia and Weismantel [4] consider the following problem:

min{ f^k (z) : z $\in$ P $\cap$ $\mathbf{Z}^n$ }, where f^k is a polynomial function of degree at most k with integer coefficients, and P is a rational polyhedron in $\mathbf{R}^n$. We recall that a rational polyhedron is the set of points that satisfy a system of linear inequalities with rational data. According to Del Pia and Weismantel [4], this problem can be solved in time polynomial for n = k = 2.

Theorem 2(Theorem 1.1 in Del Pia and Weismantel [4]). *If n = k = 2, problem min{ f^k (z) : z $\in$ P $\cap$ $\mathbf{Z}^n$ } can be solved in polynomial time.*

Recall that Theorem 2 is given(Theorem 1.1) in generalized form in aforementioned Del Pia et al. [5] as well as the following standard definitions are clearly mentioned there.

For a rational polyhedron P := {x $\in$ $\mathbf{R}^n$: $\mathbf{A}$x $\le$ $\mathbf{b}$}, with $\mathbf{A} \in \mathbf{Z}^{mxn}$, $\mathbf{b} \in \mathbf{Z}^m$ the following is defined in Del Pia et al. [5]: "...We use the words size and b-inary encoding length synonymously. The size of P is the sum of the sizes of **A** and **b**. We say that problem can be solved in polynomial time if in time b-ounded by a polynomial in the size of **A**, **b** and M we can either determine t-hat the problem is infeasible, find a feasible minimizer...". (M = 1 in our ca-se). We use here exactly the same definitions. We emphasize that according to Theorem 2, for a general rational polyhedron, the only conditions for the polynomial-time minimization are the following conditions: "n" and "k" mu-st be fixed and n = k = 2: the number of linear inequalities, "m", is not supp-osed to be fixed to provide the fact of polynomiality in time and "m" doesn't belong to the binary encoded input: it is unary encoded.

We are going now to reformulate the original problem (1) by replacing it with the equivalent problem, having the same target function, but feasible set as the integer points in some two-dimensional rational polyhedron(polygon), which therefore would be solved in polynomial time according to Theorem 2 (Theorem 1.1 in Del Pia and Weismantel [4]).

Let us construct the corresponding polyhedron G, as having the edges M_iM_{i+1}, where the vertex M_i is a point on the portion xy = N of the boundary of region Ω of (1), the point, corresponding to x = i, $2 \le i \le N - 2$, so M_i := (i, N/i), plus edges M_2A and M_{N-1}A, along two other portions(parallel to the x axis and y axis correspondingly) of three portions of the boundary of region Ω, where the vertex A := (N – 1, N/2). Polyhedron G can be described as a set of points that satisfy the corresponding system of linear inequalities with rational data, each inequality corresponds to one edge of G and can be described in the form: $x + a_i y \le b_i$, wherein $a_i = -(i + 1)i$, $b_i = i(1 - N) - N$, $2 \le i \le N - 2$, and wherein $(x, y) \in \mathbf{R}^2$, plus inequalities for edges M_2A and $M_{N-1}A$.

Discrete nature of the problem provides the following advantage.

Theorem 3. $\Omega \cap \boldsymbol{Z}^2 = G \cap \boldsymbol{Z}^2$.

Proof. It follows from definitions of Ω and G and their convexity and convexity of G follows from the convexity of Ω. □

Theorem 4. *Problem (1) is equivalent to the problem:*

$$\min\{ xy : (x,y) \in G \cap \boldsymbol{Z}^2 \} \tag{2}$$

Proof. It follows from Theorem 3 and problems (1) and (2). □

Theorem 5(Minimum Principle). *If* N *is not a prime, any minimizer of (2) is a solution of integer factorization problem for* N *and any solution of integer factorization problem for* N *is a minimizer of (2).*

Proof. It follows from Theorem 1 and Theorem 4. □

Note that rational polyhedron G can be constructed e.g. so that it contains edge M_2M_{N-1} instead of edges M_2A and $M_{N-1}A$.

Recall that the fact of polynomiality in Theorem 2 does not require that " m"(the number of inequalities) must be fixed: just "n" and "k" must be fixed in Theorem 2, wherein "m", "n" and "k" are unary encoded.

Problem (2) completely satisfies Theorem 2 (Theorem 1.1 in Del Pia and Weismantel [4]), because target function of (2) is a quadratic polynomial with integer coefficients, G is a two-dimensional rational polyhedron, and, therefore, (2), (1) and integer factorization problem would be solved in time polynomial, according to the Theorem 2 (Theorem 1.1 in Del Pia and Weismantel [4]). It means, according to aforementioned definitions that it would be solved in time, bounded by a polynomial in the size of **A** and **b**. In fact, as it was mentioned above, according to the clear definition, given in Del Pia et al . [5]: "...We say that problem can be solved in polynomial time if in time bounded by a polynomial in the size of **A**, **b** we can either determine that the problem is infeasible, find a feasible minimizer...". Thus, the fact of polynomiality in time of problem (2) means that it can be solved in time bounded by a polynomial in the size of coefficients of the inequalities, describing our poyhedron G, and according to the Theorem 2(Theorem 1.1 in Del Pia and Weismantel [4], Theorem 1.1 in Del Pia et al. [5]), this is the case(it is polynomial in time). As a result, problems (2), (1) can be solved in time bounded by a polynomial in the size of coefficients of the inequalities, describing our polyhedron G . Thus, polynomiality in time of (2) and (1) is guaranteed by Theorem 2($n = k = 2$ in our case), Theorem 4, aforementioned standard definitions and by the encoding unarity of the "m". It is important to note that since $m = N - 3$, those running time, bounded by a polynomial, comprises unary encoding, depended on N, parameter $m = N - 3$ and binary encoding length, depended on N as well. The following example demonstrates a fixed-dimensional algorithm, that can be done in time polynomial in unary variables, including "m", as well as in the binary encoding length. In fact, for aforementioned in section 2 quasiconvex polynomial integer minimization problem, similarly, it can be solved in time polynomial in the degrees and the binary encoding of the coefficients, when the dimension is fixed, as well as in "m" (in in the number of constraints), see e,g., Theorem 1.5 in Lee [9], Heinz [6], section 3.1, Theorem 10 in Hemmecke et al. [7]. In another example, again the corresponding algorithm is polynomial in "m" (in the number of constraints) and in the binary encoding of the coefficients, see, e.g., section 2.1, Theorem 5 in Hemmecke et al. [7]. In both examples, the degrees and the number of constraints are unary encoded and are not fixed.

Thus, we obtain the following algorithm:

Algorithm 2(Integer Factorization Algorithm).
Input: A positive integer number N.
Output: A nontrivial divisor of N(if it exists).

Solve the problem (2) using algorithms [4]:
Based on the input data compute
a minimizer (x_{min}, y_{min})
of the problem (2).
if $(x_{min} y_{min} = N)$
then
Return a nontrivial divisor x_{min} **of N**
else

Return "N is a prime"

Now we are going to make final conclusions about the complexity of Algorithm 2.

Three fundamental facts, considered above in full details would lead to the fact of polynomiality of the Algorithm 2.

First, as we mentioned above, according to the standard definition the fact of polynomiality in time of problem (2) means that it can be solved in time bounded by a polynomial in the size of coefficients of the inequalities, describing our polyhedron G and according to the Theorem 2 (Theorem 1.1 in Del Pia and Weismantel [4], Theorem 1.1 in Del Pia et al. [5]) this is the case: it is polynomial in time.

Second, two examples, described above in full details, demonstrate a role of unary encoded unfixed parameters, which provide, nevertheless, algorithms that are not exponential, they are polynomial.

Third, all coefficients of the inequalities, describing our polyhedron G are polynomial integer functions of N(Recall them: $x + a_i y \le b_i$, wherein $a_i = -(i + 1)i$, $b_i = i(1 - N) - N$, $2 \le i \le N - 2$, and wherein $(x, y) \in \mathbf{R}^2$, plus inequalities for edges M_2A and $M_{N-1}A$) of the degree, not greater than 2.

However, since the fact of polynomiality in time of problem (2) means that it can be solved in time, bounded by a polynomial in the size of G, so in the sum of sizes of **A** and **b**, Algorithm 2 does not run in time polynomial in log(N).

Let us develop another integer factorization algorithms that use our rational polyhedron G, constructed above by us.

Note that any solution of integer factorization problem for a non-prime N corresponds to the certain vertex M := (p, d) of G, where both p and d are integers.

(Here and further we use rational polyhedron G that contains edge M_2M_{N-1} instead of edges M_2A and $M_{N-1}A$).

We will use the following Theorem 6(aforementioned section 2.1, Theorem 5 in Hemmecke et al. [7], by Cook et al.).

Theorem 6. *Let $P = \{x \in \boldsymbol{R}^n : \boldsymbol{A}x \leq \boldsymbol{b}\}$ be a rational polyhedron with $\boldsymbol{A} \in \boldsymbol{Q}^{mxn}$ and let φ be the largest binary encoding size of any of the rows of the system $\boldsymbol{A}x \leq \boldsymbol{b}$. Let $P^I = conv(P \cap \boldsymbol{Z}^n)$ be the integer hull of P. Then the number of vertices of P^I is at most $2m^n(6\, n^2\varphi)^{n-1}$.*

Let us apply Theorem 6 to our rational polyhedron G(Let P : = G).

Due to convexity of G its clear that all vertices of G corresponding to solutions of integer factorization problem for a non-prime N belong to the set of vertices of the integer hull of G and according to the mentioned above section 2.1, Theorem 5 in Hemmecke et al. [7]: "... when the dimension is fixed there is only a polynomial number of vertices ... ". On the other hand, as its mentioned in those section 2.1 in Hemmecke et al. [7]: "... Moreover, Hartmann [64] gave an algorithm for enumerating all the vertices, which runs in polynomial time in fixed dimension...". That is why by applying aforementioned Hartmann's algorithm for enumeration of the vertices of the integer hull of our polyhedron G, we get a polynomial-time algorithm for integer factorization, polynomial in log(N), because in contrast to the aforementioned theory, described in Del Pia and Weismantel [4], Del Pia et al. [5], input size considered here, according to the Theorem 6 is "... the largest binary encoding size of any of the rows of the system **A**x ≤ **b** ...", not "... the sum of sizes of **A** and **b**", as its defined in Del Pia and Weismantel [4], Del Pia et al. [5].

Algorithm 3(Integer Factorization Algorithm).
Input: A positive integer number N.
Output: A nontrivial divisor of N(if it exists).

```
while(next vertex)
{
Enumerate vertices of the corresponding
integer hull of the polyhedron G by using
Hartmann's algorithm and when a
vertex (p, d) is enumerated, issue verification:
if (pd  =  N)
    Return a nontrivial divisor d of N
}
    Return "N is a prime"
```

So, the key fact, leading to solution, is the definition of binary input, given in the Theorem 6 as: "... the largest binary encoding size of any of the rows of the system **A**x ≤ **b** ..." together with aforementioned three fundamental facts.

That is why the fact of polynomiality is preserved this time and Algorithm 3 runs in time polynomial in log(N) as well.

Thus, factoring is in FP. The class FP is the set of function problems which can be solved by a deterministic Turing machine in polynomial time (see, e.g., Cormen et al. [3]).

Theorem 7. *Integer factorization is in FP.*

Algorithm 3 can be modified to serve the decision problem version as well - given an integer N and an integer q with $1 \leq q \leq N$, does N have a factor d with $1 < d < q$?

Let $\Omega_q := \{ (x, y) \in \mathbf{R}^2 \mid xy \geq N, \ 2 \leq x \leq q - 1, \ N/(q - 1) \leq y \leq N/2, \ x \in \mathbf{R}, \ y \in \mathbf{R} \}$ for a given q, $3 \leq q \leq N$, $N \in \mathbf{N}$.

Let G_q rational polyhedron like G, but corresponding to Ω_q.

Algorithm 4(Integer Factorization Algorithm).

Input: A positive integer numbers N, $q < N$.

Output: Existence of a factor d with $1 < d < q$.

```
while(next vertex)
{
Enumerate vertices of the corresponding
integer hull of the polyhedron Gq by using
Hartmann's algorithm and when a
vertex (p, d) is enumerated, issue verification:
if (pd   =   N)
    Return  "The corresponding factor exists"
}
    Return  "The corresponding factor does not exist"
```

Hence, Algorithm 4 runs in time polynomial in log(N) as well.

Thus, factoring is in P. The class P is the class of sets accepted by a deterministic polynomial-time Turing machines (see, e.g., Cormen et al. [3]).

Theorem 8. *Integer factorization is in P.*

Note that algorithms 3 – 4 can be considered as polynomial-time primality tests and the only provably polynomial-time primality test was developed by Agrawal et al. [1].

Theorem 9. $P = NP$.

Proof. It is well known that if there is a polynomial-time algorithm for any NP-hard problem, then there are polynomial-time algorithms for all proble-ms in NP, and hence, we would conclude that P is equal NP. The class NP is the set of decision problems solvable in polynomial time by a theoretical non-deterministic Turing machine(see, e.g., Cormen et al. [3]). We get the proof of Theorem 8 due to the Theorem 1(Minimum Principle), NP-hardness of (1) and Theorem 7. □

References

[1] M. Agrawal, N. Kayal, N. Saxena, PRIMES is in P, Annals of Mathema-tics 160(2) (2004) 781–793.

[2] M. Baes, T. Oertel, C. Wagner, R. Weismantel, Mirror-Descent Methods in Mixed-Integer Convex Optimization, in: M. Jünger, G. Reinelt (Eds.), Facets of combinatorial optimization, Springer, Berlin, New York, 2013, pp. 101–131, available electronically from http://arxiv.org/pdf/1209.0686.pdf

[3] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction To Algorith-ms, third ed, The MIT Press, Cambridge, 2009.

[4] A. Del Pia, R. Weismantel, Integer quadratic programming in the plane, Proceedings of SODA, 2014, pp. 840-846, available electronically from https://sites.google.com/site/albertodelpia/home/publications

[5] A. Del Pia, R. Hildebrand, R. Weismantel, K. Zemmer, Minimizing Cu-bic and Homogeneous Polynomials over Integers in the Plane,To appear in Mathematics of Operations Research (2015), available electronically from https://arxiv.org/pdf/1408.4711.pdf

[6] S. Heinz, Complexity of integer quasiconvex polynomial optimization, J. Complexity 21(4) (2005) 543–556.

[7] R. Hemmecke, M. Köppe, J. Lee, R. Weismantel, Nonlinear Integer Pro-gramming, in: M. Jünger, T. Liebling, D. Naddef, W. Pulleyblank, G. Reinelt, G. Rinaldi, L.Wolsey (Eds.), 50 Years of Integer Programming 1958–2008: The Early Years and State-of-the-Art Surveys, Springer-Ve-rlag, Berlin, 2010, pp. 561–618, available electronically from

http://arxiv.org/pdf/0906.5171.pdf
[8] L. G. Khachiyan, L. Porkolab, Integer optimization on convex semialgebraic sets, Discrete and Computational Geometry 23(2) (2000) 207–224.
[9] J. Lee, On the boundary of tractability for nonlinear discrete optimization, in: Cologne Twente Workshop 2009, 8th Cologne Twente Workshop on Graphs and Combinatorial Optimization, Ecole Polytechnique, Paris, 2009, pp. 374–383, available electronically from http://www.lix.polytechnique.fr/ctw09/ctw09-proceedings.pdf#page=385
[10] A. K. Lenstra, H. W. Jr. Lenstra, (Eds.), The development of the number field sieve, Springer-Verlag, Berlin, 1993.
[11] T. Oertel, C. Wagner, R. Weismantel, Convex integer minimization in fixed dimension, CoRR 1203–4175(2012), available electronically from http://arxiv.org/pdf/1203.4175.pdf
[12] P. Stevenhagen, The number field sieve, Algorithmic Number Theory: Lattices, Number Fields, Curves, and Cryptography, Mathematical Sciences Research Institute Publications, Cambridge University Press, Cambridge, 2008.