P=NP via Integer Factorization and Optimization

Yuly Shipilevsky

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

Abstract

We develop two different polynomial-time integer factorization algorithms.

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

Next, we reduce those minimization problem to the polynomial-time minimizing a quadratic polynomial with integer coefficients over the integer points in a special two-dimensional rational polyhedron.

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 show that there exists an NP-hard minimization problem, equivalent to the original minimization 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 [11], P. Stevenhagen [13]).

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. [4]).

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 analyze complexity of obtained integer minimization problem.

In Section 3 we consider equivalent minimization problems with transformed target functions.

In Section 4 we construct a special two-dimensional rational polyhedron and reduce those integer minimization problem to the integer minimization problem over the integer points in that rational polyhedron and show that it can be solved in time polynomial.

We develop a polynomial-time algorithm for integer factorization by enumeration of vertices of integer hull of that two-dimensional rational polyhedron using M. Hartmann's algorithm and selection of the integer points.

In Section 5 we show that there exists an NP-hard problem, equivalent to to the original two-dimensional integer minimization problem and since the original problem is equivalent to integer factorization, which is in P, we conclude that P = NP.

In Section 6 we make conclusions.

2. Reduction to the Integer Programming problem. Minimum Principle. Equivalence

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:

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

$$x \in \mathbf{N},\ y \in \mathbf{N},\ N \in \mathbf{N}.$$

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 problem, that solution is a minimizer of problem (1) and if there exists a nontrivial solution of integer factorization problem, any minimizer of problem (1) is a nontrivial solution of integer factorization.*

Corollary 1(Equivalence). *Problem (1) and integer factorization are equivalent.*

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. [8]), 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 [9]).

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(see, e.g., Heinz [7], Hemmecke et al. [8], Lee [10]). 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], Oertel et al. [12].

Since the objective function $f(x, y) = xy$ of the problem (1) is a quasiconcave function in the feasible set Ω of the problem (1), we cannot use the results described in Baes et al. [2], Heinz [7], Hemmecke et al. [8], Khachiyan and Porkolab [9], Oertel et al. [12] in order to solve the problem (1) in time polynomial in log(N). Note that Ω^I is described by quasiconvex polynomials, since $(-xy + N)$ is a quasiconvex function for $x > 0$, $y > 0$.

The epigraph form(see e.g., Boyd and Vandenberghe [3], section 4.2.4) of the original problem (1) includes a new non-convex constraint, so it does not improve complexity as well.

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. [8]), but rather are NP-hard, since the problem (1) is a two-dimensional non-convex quadratically constrained quadratic integer minimization problem(see, e.g., Del Pia and Weismantel [5], Del Pia et al. [6]).

3. U-equivalent minimization

The following results give us a possibility to change the properties of the objective function with preservation of the set of minimizers of the original problem.

Recall that a function U: $\mathbf{R} \to \mathbf{R}$, $U = U(u)$ is called strictly increasing, if for all $u_1 \in \mathbf{R}$ and $u_2 \in \mathbf{R}$ such that $u_1 < u_2$ one has $U(u_1) < U(u_2)$.

We give a comprehensive proof of the fact that a conversion of the target function using any monotonic strictly increasing function U: $\mathbf{R} \to \mathbf{R}$, $U = U(u)$ preserves the original set of minimizers. The idea itself, is given, e.g., in Boyd and Vandenberghe [3], section 4.1.3.

Theorem 2. *Let* O *be the minimization problem:*

$$O = \{\textit{minimize } g(x) \textit{ subject to } x \in G\},\ g: X \to \mathbf{R},\ G \subseteq X.$$

Let E *be the minimization problem:*

$$E = \{\textit{minimize } U(g(x)) \textit{ subject to } x \in G\},\ G \subseteq X,$$

where U: $\mathbf{R} \to \mathbf{R}$, $U = U(u)$ *is any strictly increasing function.*

Let M_O *be a set of minimizers of problem* O *and*

let M_E *be a set of minimizers of problem* E.

Then:

$$M_O = M_E \ (argmin(O) = argmin(E)) .$$

Proof. If $x_0 \in M_O$ then $g(x_0) \le g(x)$ for any $x \in G$. Hence, $U(g(x_0)) \le U(g(x))$ for any $x \in G$, since function U is strictly increasing function, and, therefore: $x_0 \in M_E$ and $M_O \subseteq M_E$. If $x_0 \in M_E$ then we have: $U(g(x_0)) \le U(g(x))$ for any $x \in G$, and, therefore: $g(x_0) \le g(x)$ for any $x \in G$, as otherwise there exists $y_0 \in G$ such that $g(x_0) > g(y_0)$ and since function U is strictly increasing function, it would mean that $U(g(x_0)) > U(g(y_0))$ in contradiction to the original supposition that $U(g(x_0)) \le U(g(x))$ for any $x \in G$. So, since $g(x_0) \le g(x)$ for any $x \in G$, then $x_0 \in M_O$ and $M_E \subseteq M_O$ and finally: $M_O = M_E$. □

Definition 1. *We say that the minimization problem:*

$$E = \{\textit{minimize}\ \ U(g(x))\ \ \textit{subject to}\ \ x \in G\}$$

is U*–equivalent to the minimization problem:*

$$O = \{\textit{minimize}\ \ g(x)\ \ \textit{subject to}\ \ x \in G\},$$

$$g:\ X \to \mathbf{R},\ G \subseteq X,$$

where $U:\ \mathbf{R} \to \mathbf{R},\ U = U(u)$ *is some strictly increasing function.*

Corollary 2. *If* E *is* U*-equivalent to* O *then* E *and* O *have the same set of minimizers: argmin(O) = argmin(E).*

Proof. It follows from Theorem 2 and Definition 1. □

Note that monotonic strictly increasing function $U:\ \mathbf{R} \to \mathbf{R}, U = U(u)$ can be not continuous.

Thus, using U-equivalence we can convert original minimization problem into minimization problem that has objective function with desired properties, so that both problems, - the original one, and U-equivalent have the same set of minimizers and share the same feasible set.

Hence, as a result of the U-equivalent conversion the original feasible set and the original set of minimizers remain unchanged, whereas the objective function is being changed to obtain desired properties (e.g., faster minimization), which can consider it(U-equivalence) as a flexible and effective tool.

U-equivalent conversion can be considered as unary operation defined on the set of minimization problems, having the same feasible set.

Example 1.

O = { Minimize ax subject to $x \geq 0,\ a > 0, x \in \mathbf{R}, a \in \mathbf{R}$ },
$U(u) = bu, ,\ b > 0, u \geq 0, u \in \mathbf{R}, b \in \mathbf{R}$,
E = { Minimize abx subject to $x \geq 0,\ a > 0, b > 0, x \in \mathbf{R}, a \in \mathbf{R}, b \in \mathbf{R}$ },
E is bu-equivalent to O,
argmin(O) = argmin(E) = 0, O = E = 0.

Example 2.

O = { Minimize ax subject to $x \geq 0$, $a > 0$, $x \in \mathbf{R}$, $a \in \mathbf{R}$ },
U(u) = bsin(u), , $b > 0$, $0 \leq u \leq \pi/2$, $u \in \mathbf{R}$, $b \in \mathbf{R}$,
E = { Minimize bsin(ax) s. t. $x \geq 0$, $a > 0$, $b > 0$, $x \in \mathbf{R}$, $a \in \mathbf{R}$, $b \in \mathbf{R}$ },
E is bsin(u)-equivalent to O,
argmin(O) = argmin(E) = 0, O = E = 0.

Example 3.

O = { Minimize $x^2 + y^2 + 1$ subject to $x \in \mathbf{R}$, $y \in \mathbf{R}$ },
U(u) = log(u), , $u > 0$, $u \in \mathbf{R}$,
E = { Minimize $\log(x^2 + y^2 + 1)$ subject to $x \in \mathbf{R}$, $y \in \mathbf{R}$ },
E is log(u)-equivalent to O,
argmin(O) = argmin(E) = (0, 0), O = 1, E = 0.

Example 4. Suppose, the problem (1) is the original minimization problem. Let q be e^u-equivalent to the problem (1). The objective function of the problem (1) is xy, whereas the objective function of q is $f(x, y) = e^{xy}$. Both problems, due to the Theorem 2 have the same set of minimizers (and each such minimizer is a solution of the integer factorization problem, according to the Theorem 1). Note that if N is not a prime, minimum $q = e^N$.

Similar example:

O = { Minimize xy subject to $xy \geq 6$, $2 \leq x \leq 5$, $3 \leq y \leq 5$, $x \in \mathbf{Z}$, $y \in \mathbf{Z}$},
$U(u) = e^u$, $u \in \mathbf{R}$,
E = { Minimize e^{xy} subject to $xy \geq 6$, $2 \leq x \leq 5$, $3 \leq y \leq 5$, $x \in \mathbf{Z}$, $y \in \mathbf{Z}$},
E is e^u-equivalent to O,
argmin(O) = argmin(E) = (2,3), O = 6, $E = e^6$,
e = 2.71828...(Euler's number).

More complicated example: a u^u-equivalent problem to the problem (1).

However, no U-equivalent conversion applied to the original problem (1) in order to get a quasiconvex objective function exists, since if a function g is quasiconcave and a function U is increasing, then a function f, defined as f (x) = U(g(x)) is still quasiconcave.

We will use U-equivalence and results obtained in Section 2 at the end of Section 5, in order to prove that P = NP.

4. Linearization. Polynomial-time Integer Factorization

It was shown in Del Pia and Weismantel [5] that problem of minimizing a quadratic polynomial with integer coefficients over the integer points in a general two-dimensional rational polyhedron is solvable in time bounded by a polynomial in the input size and it was further extended to cubic and homogeneous polynomials in Del Pia et al. [6].

Del Pia and Weismantel [5] 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 [5], this problem can be solved in time polynomial for n = k = 2.

Theorem 3(Theorem 1.1 in Del Pia and Weismantel [5]). *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 3 is given(Theorem 1.1) in generalized form in aforementioned Del Pia et al. [6] as well as the following standard definitions are clearly mentioned there.

For a rational polyhedron $P := \{x \in \mathbf{R}^n : \mathbf{A}x \leq \mathbf{b}\}$, with $\mathbf{A} \in \mathbf{Z}^{mxn}$, $\mathbf{b} \in \mathbf{Z}^m$ the following is defined in Del Pia et al. [6]: "...We use the words size and binary 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 bounded by a polynomial in the size of **A**, **b** and M we can either determine that the problem is infeasible, find a feasible minimizer...". (M = 1 in our case). We use here exactly the same definitions. We emphasize that according to Theorem 3, for a general rational polyhedron, the only conditions for the polynomial-time minimization are the following conditions: "n" and "k" must be fixed and n = k = 2: the number of linear inequalities, "m" (the number of facets of P), is not supposed 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 3 (Theorem 1.1 in Del Pia and Weismantel [5]).

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 $M_{N-1}A$. Thus, m = N – 3 + 2 = N – 1.

Discrete nature of the problem provides the following advantage.

Theorem 4. $\Omega \cap \boldsymbol{Z}^2 = G \cap \boldsymbol{Z}^2$.

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

Theorem 5. *Problem (1) is equivalent to the problem:*

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

Proof. It follows from Theorem 4 and problems (1) and (2). □

Theorem 6(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 5. □

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 3 does not require that " m"(the number of inequalities) must be fixed: just "n" and "k" must be fixed in Theorem 3, wherein "m", "n" and "k" are unary encoded.

Problem (2) completely satisfies Theorem 3 (Theorem 1.1 in Del Pia and Weismantel [5]), 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 3 (Theorem 1.1 in Del Pia and Weismantel [5]). 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 . [6]: "...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 3(Theorem 1.1 in Del Pia and Weismantel [5], Theorem 1.1 in Del Pia et al. [6]), 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 3(n = k =2 in our case), Theorem 5, aforementioned standard definitions and by the encoding unarity of the "m". It is important to note that since m = N – 1, those running time, bounded by a polynomial, comprises unary encoding, depended on N, parameter m = N – 1 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 the number of constraints, see, e,g., Theorem 1.5 in Lee [10], Heinz [7], section 3.1, Theorem 10 in Hemmecke et al. [8]. 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. [8].

In both examples, the degrees and the number of constraints are unary encoded and are not fixed, nevertheless, they can be solved in time polynomial.

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 [5]:
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 3 (Theorem 1.1 in Del Pia and Weismantel [5], Theorem 1.1 in Del Pia et al. [6]) this is the case: it is polynomial in time.

Thus, binary input only is important in making a decision about the fact of polynomiality.

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 $a_i = -(i + 1)i$, $b_i = i(1 - N) - N$, $2 \le i \le N - 2$, $(x, y) \in \mathbf{R^2}$, plus inequalities for edges M_2A and $M_{N-1}A$) of the degree, not greater than two.

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**(according to the definition, given in Del Pia et al. [6]), Algorithm 2 does not run in time polynomial in log(N).

However, it is well known to define the size of rational polyhedron, described as $G := \{x \in \mathbf{R}^n : \mathbf{A}x \le \mathbf{b}\}$ as the largest binary encoding size of any of the rows of the system $\mathbf{A}x \le \mathbf{b}$ (see, e.g., section 2.1, Theorem 5 in Hemmecke et al. [8]). It is clear that under such definition, Algorithm 2 runs in time polynomial in log(N) as well, because each such row in our case consist of 2 coefficients of the corresponding matrix **A** and one coefficient of **b**, each coefficient is a polynomial integer functions of N of the degree, not greater than two(see aforementioned third fundamental fact).

Algorithm 2 can be modified to serve the decision problem version as well - given an integer N and an integer q with $1 \le q \le N$, does N have a factor d with $1 < d < q$?

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

Let G_q rational polyhedron, corresponding to Ω_q. Let $G_q^I := G_q \cap \mathbf{Z^2}$.

Let us replace (2) by the problem over the feasible set G_q^I and denote the modified minimization problem (corresponding to the problem (2)) as (3).

Algorithm 3(Integer Factorization Algorithm).
Input: Positive integer numbers N, $q < N$.
Output: Existence of a factor d with $1 < d < q$.

Solve the problem (3) using algorithms [5]:
Based on the input data compute
a minimizer (x_{min}, y_{min})
of the problem (3)
if $(x_{min} y_{min} = N)$
then
Return "The corresponding factor exists"
else

Return "The corresponding factor does not exist"

Hence, Algorithm 3 runs in time polynomial in log(N) as well.

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 $M_{N-1}A$, so it has inequality description: $c_ix + a_i y \leq b_i$, wherein $c_i = 1$, $a_i = -(i + 1)i$, $b_i = i(1 - N) - N$, $2 \leq i \leq N - 2$, $c_{N-1} = N$, $a_{N-1} = 2(N - 1)$, $b_{N-1} = N(N + 1)$, $(x, y) \in \mathbf{R^2}$.

We will use the following Theorem 7(aforementioned section 2.1, Theorem 5 in Hemmecke et al. [8]: "... when the dimension is fixed, there is only a polynomial number of vertices, as Cook et al. [38] showed ...").

Theorem 7. *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 $2m^n(6\, n^2\varphi)^{n-1}$.*

Note that integer hull P^I is a polyhedron: see aforementioned section 2.1 in Hemmecke et al. [8]: " ... Maximizing a convex function over the integer points in a polytope in fixed dimension can be done in polynomial time. To see this, note that the optimal value is taken on at a vertex of the convex hull of all feasible integer points...". Note as well that all the vertices of P^I are the integer points.

Let us apply Theorem 7 to our rational polyhedron G(Let P : = G, n = 2, m = N – 3 + 1 = N – 2).

According to it's definition, each vertex (p, d) of the set of vertices V(G) of our rational polyhedron G includes rational p and rational d, and if N is not a prime, some of them, corresponding to the solution of integer factorization problem, includes both integer p and integer d: the integer points.

Due to convexity of our rational polyhedron(polygon) G, its clear that all the vertices $V^I(G)$ of G, corresponding to the solution of integer factorization problem for a non-prime N, since they all belong to the boundary of convex polygon G and since $G^I \subseteq G$, they all belong to the set of vertices $V(G^I)$ of the integer hull G^I of G, and regarding complexity of $V(G^I)$, according to the mentioned above section 2.1 in Hemmecke et al. [8]: "... when the dimension is fixed there is only a polynomial number of vertices, as Cook et al. [38] showed...". As a result, we get the following Theorem 8.

Theorem 8. $V^I(G) \subset V(G^I)$.

On the other hand, as its mentioned in those section 2.1 in Hemmecke et al. [8]: "... Moreover, Hartmann [64] gave an algorithm for enumerating all the vertices, which runs in polynomial time in fixed dimension...".

That is why, due to Theorems 7 and 8, by applying aforementioned Hartmann's algorithm for enumeration of the vertices $V(G^I)$ of the integer hull of our polyhedron G and further selection of points, satisfying integer factorization condition, we get a polynomial-time algorithm for integer factorization, polynomial in log(N), since similar to the aforementioned theory, described in Del Pia and Weismantel [5], Del Pia et al. [6], the input size considered here, according to the Theorem 7 is "... the largest binary encoding size of any of the rows of the system $\mathbf{A}x \leq \mathbf{b}$ ...", not "... the sum of sizes of **A** and **b**", as its defined in Del Pia and Weismantel [5], Del Pia et al. [6].

Algorithm 4(Integer Factorization Algorithm).
Input: A positive integer number N.
Output: A nontrivial divisor of N(if it exists).

```
while(next vertex)
{
Enumerate vertices V(G^I ) of the corresponding
integer hull G^I 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"
```

For clear understanding of Algorithm 4 we strongly recommend to understand all the details given in aforementioned Cook et al. and Hartmann's papers.

So, the key fact, leading to solution, is the definition of binary input, given in the Theorem 7 as: "... the largest binary encoding size of any of the rows of the system $\mathbf{A}x \leq \mathbf{b}$ ..." together with aforementioned three fundamental facts.

Recall that according to the standard definition, the fact of polynomiality means, that the problem can be solved in time, bounded by a polynomial in the size of coefficients of the inequalities, describing our polyhedron G, so according to the binary input size only.

That is why the fact of polynomiality is preserved this time as well and the Algorithm 4 runs in time polynomial in log(N) as well.

Ones again, its extremely important to understand that "N" is in use in the coefficients of the system $\mathbf{A}x \leq \mathbf{b}$, describing our rational polyhedron G, and, on the other hand, "N" means the number of facets of G (the number of inequalities of the system $\mathbf{A}x \leq \mathbf{b}$). As a number of facets of G, "N" is linearly transformed into "m" parameter ("dressed into clothes" of "m", is inside the "m"), "m" is unfixed unary (not a binary!) encoded and the complexity of both our algorithms is polynomial-time in unary encoded "m".

So "N" facets must be treated here as unary encoded "m", and, again, the complexity is polynomial in "m" as its clearly stated in aforementioned papers.

Recall again, that binary input only is important in making a decision about the fact of polynomiality in time.

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. [4]).

Theorem 9. *Integer factorization is in FP.*

Algorithm 4 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 5(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 G_q 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 5 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. [4]).

Theorem 10. *Integer factorization is in P.*

Note that algorithms 2 – 5 can be considered as polynomial-time primality tests and the only provably polynomial-time primality test was developed by Agrawal et al. [1].

5. NP-hard Equivalence. P = NP

Theorem 11. *P = NP.*

Proof. Recall that problem (1) is a two-dimensional, non-convex, quadratically constrained quadratic integer minimization problem(see, e.g., Del Pia and Weismantel [5], Del Pia et al. [6]).

If problem (1) is NP-hard, then due to the Theorem 1(Minimum Principle) , Corollary 1, it is equivalent to the integer factorization, which is in P, according to the Theorem 10. Therefore, P = NP, since if there is a polynomial-time algorithm for any NP-hard problem then there are polynomial-time algorithms for all problems in NP.

Let us suppose that problem (1) is not an NP-hard problem.

Note that according to Hemmecke et al. [8]: "... as soon as we add just two integer variables, we get a hard problem again: Theorem 2. The problem of minimizing a degree-4 polynomial over the lattice points of a convex polygon is NP-hard... This is based on the NP-completeness of the problem whether there exists a positive integer $x < c$ with $x^2 \equiv a \pmod{b}$; see [53, 41] ..." and note that according to Del Pia et al. [6]: "... Using the same reduction as Lemma 1.2, it is possible to show that problem (1) is NP-hard even when $n = d = 2$, P is a bounded, rational polyhedron, and we add a single quadratic inequality constraint (see [18]) ...".

These two fundamental facts would provide a ground to use a degree of target integer polynomials as a parameter, causing guaranteed NP-hardness of at least one of considering two-dimensional, non-convex integer minimization problems(therefore, P = NP, since if there is a polynomial-time algorithm for any NP-hard problem then there are polynomial-time algorithms for all problems in NP).

Note that of course, it doesn't mean that any quartic minimization problem or problem with a larger exponent is also NP-hard, and it doesn't mean that any quadratic optimization with one quadratic constraint is NP-hard.

We are going to use U-equivalences' methods, described in Section 3.

Let us consider the following class of polynomial functions with integer coefficients: $P_n(u) := a_n u^n + a_{n-1}u^{n-1} + ... + a_1 u + a_0$, $a_i \in \mathbf{Z}$, $u > 0$, $u \in \mathbf{R}$, $n \in \mathbf{N}$. Suppose that $a_i \in \mathbf{Z}$ are chosen so that $P_n(u)$ is a monotone, strictly increasing function. In trivial cases, all $a_i \geq 0$.

There exists such a number, $n = n_0 \geq 2$ that some the following $P_n(u)$-equ-

ivalent problem (see Section 3, Corollary 2) to the original problem (1) is NP-hard (otherwise, for all $n \geq 2$, any such problem is not NP-hard, which cannot be true for unbounded $n \in \mathbf{N}$, due to aforementioned two fundamental facts):

$$\text{minimize } P_n(u), \; u = xy,$$

$$\text{subject to } (x,y) \in \Omega^I.$$

In simplest case: $P_n(u) := u^n$, $u = xy$, . $n \in \mathbf{N}$.

Thus, if problem (1) is NP-hard, then due to the Theorem 1 (Minimum Principle), Corollary 1, it is equivalent to the integer factorization, which is in P, according to the Theorem 10. Therefore, P = NP, since if there is a polynomial-time algorithm for any NP-hard problem, then there are polynomial-time algorithms for all problems in NP.

If problem (1) is not NP-hard, then for some $n = n_0 \geq 2$, there exists some NP-hard problem, P_n (u)-equivalent to the problem (1), which in its turn, due to the Theorem 1(Minimum Principle), Corollary 1, is equivalent to the integer factorization, which is in P, according to the Theorem 10. Therefore, P=NP, since, again, if there is a polynomial-time algorithm for any NP-hard problem, then there are polynomial-time algorithms for all problems in NP.

More complicated U-equivalences to the original problem (1):

a. A non-convex, non-polynomial target of e^u-equivalence, considered in Section 3, Example 4.
b. A non-convex, non-polynomial target of P_n (u^u)-equivalence.

It is clear that we could construct unlimited number of more and more sophisticated non-convex, non-polynomial problems, U-equivalent to the original problem (1). For example:

$$\text{minimize } e^u, \; u = (xy)^z, \; z = (xy)^v, \; v = xy,$$

$$\text{subject to } (x,y) \in \Omega^I.$$

Again, since for sure at least one such U-equivalent problem is NP-hard then P = NP, since if there is a polynomial-time algorithm for any NP-hard problem then there are polynomial-time algorithms for all problems in NP.□

6. Conclusions

We developed two different polynomial-time algorithms for integer factorization.

We associated the problem of factorization of integer number N with a rational polygon, given as a list of vertices:

$M_i := (i, N/i),\ 2 \le i \le N - 1.$

We considered the corresponding inequality description of that polygon: $c_i x + a_i y \le b_i$, wherein $c_i = 1$, $a_i = -(i + 1)i$, $b_i = i(1 - N) - N$, $2 \le i \le N - 2$, $c_{N-1} = N$, $a_{N-1} = 2(N - 1)$, $b_{N-1} = N(N + 1)$, $(x, y) \in \mathbf{R}^2$ and applied a polynomial-time Hartmann's algorithm for enumeration of vertices of the corresponding integer hull of that polygon and further testing them for satisfiability to the integer factorization of N.

We reduced integer factorization problem to equivalent problem of minimizing a quadratic polynomial with integer coefficients over the integer points in a quadratically constrained two-dimensional region and then to the polynomial-time minimization over the integer points in the corresponding, specially constructed aforementioned rational polyhedron M_i.

Since either those original minimization problem or U-equivalent problem is NP-hard problem we concluded that P = NP.

References

[1] M. Agrawal, N. Kayal, N. Saxena, PRIMES is in P, Annals of Mathematics 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. http://arxiv.org/pdf/1209.0686.pdf

[3] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf
[4] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction To Algorithms, third ed, The MIT Press, Cambridge, 2009.
[5] A. Del Pia, R. Weismantel, Integer quadratic programming in the plane, Proceedings of SODA, 2014, pp. 840-846.
https://sites.google.com/site/albertodelpia/home/publications
[6] A. Del Pia, R. Hildebrand, R. Weismantel, K. Zemmer, Minimizing Cubic and Homogeneous Polynomials over Integers in the Plane,To appear in Mathematics of Operations Research (2015).
https://arxiv.org/pdf/1408.4711.pdf
[7] S. Heinz, Complexity of integer quasiconvex polynomial optimization, J. Complexity 21(4) (2005) 543–556.
[8] R. Hemmecke, M. Köppe, J. Lee, R. Weismantel, Nonlinear Integer Programming, 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-Verlag, Berlin, 2010, pp. 561–618. http://arxiv.org/pdf/0906.5171.pdf
[9] L. G. Khachiyan, L. Porkolab, Integer optimization on convex semialgebraic sets, Discrete and Computational Geometry 23(2) (2000) 207–224.
[10] 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.
http://www.lix.polytechnique.fr/ctw09/ctw09-proceedings.pdf#page=385
[11] A. K. Lenstra, H. W. Jr. Lenstra, (Eds.), The development of the number field sieve, Springer-Verlag, Berlin, 1993.
[12] T. Oertel, C. Wagner, R. Weismantel, Convex integer minimization in fixed dimension, CoRR 1203–4175(2012).
http://arxiv.org/pdf/1203.4175.pdf
[13] 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.