



Some Considerations in Favor of the Truth of Goldbach's Conjecture

C. D'Urso

LUMSA University, Rome, Italy

Information Systems Development Office, ICT Department, Italian Senate, Rome, Italy

Abstract. This article presents some considerations about the Goldbach's conjecture (GC). The work is based on analytic results of the number theory and it provides a constructive method that permits, given an even integer, to find at least a pair of prime numbers according to the GC. It will be shown how the method can be implemented by an algorithm coded in a high-level language for numerical computation. Eventually a correlation will be provided between this constructive method and a class of problems of operations research.

2010 Mathematics Subject Classifications: 11P32, 90B99

Key Words and Phrases: Goldbach's Conjecture, Congruencies, Chinese remainder theorem, Operations research, Convex feasibility problem, Knapsack problem, Constraint satisfaction problem, Primal-dual test

1. Introduction

In his famous letter to Leonhard Euler dated 7 June 1742 [11], Christian Goldbach stated his famous conjecture that every integer greater than 2 can be written as the sum of three primes, statement that is equivalent to that every even number is a sum of two primes, as Euler stated in the letter dated 30 June 1742 ("I regard this as a completely certain theorem, although I cannot prove it"). This strong Goldbach conjecture implies the conjecture that all odd numbers greater than 7 are the sum of three odd primes, which is known today variously as the "weak" Goldbach conjecture, the "odd" Goldbach conjecture, or the "ternary" Goldbach conjecture.

In 1923 the ternary conjecture has been proved under the assumption of the truth of the generalized Riemann hypothesis [13], in 1937 Vinogradov [20] removed the dependence on the Riemann Hypothesis, and proved that this is true for all sufficiently large odd integers. In 1956 Borodzkin showed that an integer greater than a large integer is sufficient in Vinogradov's proof. In 1989 and 1996 Chen and Wang reduced this bound [6]. Using Vinogradov's

Email address: cirodurso@ieee.org

method, Chudakov, van der Corput, and Estermann [7, 19, 8] showed that almost all even numbers can be written as the sum of two primes (in the sense that the fraction of even numbers which can be so written tends towards 1). In a very recent paper, mathematician Terence Tao of the University of California, Los Angeles, has inched toward a proof, in fact he has shown that one can write odd numbers as sums of, at most, five primes [18]. Although believed to be true, the binary Goldbach conjecture is still lacking a proof. We state it as follows.

Statement 1. *Every even integer greater than 4 can be written as the sum of two primes.*

2. Alternative Statements of the Problem

Consider an even natural number $2e$, we can state the conjecture in term of the natural number e as follows.

Statement 2. *For every integer $e > 3$ there exists a couple of odd prime numbers q_1 and q_2 such that e is their semi-sum, that is: $e = \frac{q_1+q_2}{2}$.*

This statement is equivalent to the original one considering that $2e$, with e a generic integer, is always an even natural, and that the original statement is trivially verified for the numbers 4 and 6.

Definition 1. *Two prime numbers q_1 and q_2 , with $q_2 > e > q_1$, are called mirror primes respect to an integer e if there exists an integer d such that: $e - q_1 = q_2 - e = d$.*

So another equivalent statement to 2, and of course to 1, is:

Statement 3. *For every natural $e > 3$ there exists a pair of mirror primes respect to e .*

In the following we will consider, without loss of generality, $e > 7$. The statement 3, leads us to analyze when, given a natural number e , there exists at least one integer d defining two mirror primes respect to e . First of all we have to ask if and when there exist such number d (if e is prime the choice is trivially $d = 0$). Moreover we are interested in an integer such that the two integers $e \pm d$ are primes (observe that the primality can be checked by means of an efficient algorithm currently available [2]).

3. Notation

Let observe that in the following we use that notation:

With the symbol $\pi(x)$ we denote the number of primes less than or equal to x .

With the expression $a \equiv b \pmod{c}$ we state that a is congruent to b modulo c .

With the expression $a \not\equiv b \pmod{c}$ we state that a is not congruent to b modulo c .

With the expression $(n, m]$ we indicate the interval open on left and close on right.

With the expression $a \nmid b$ we denote that a doesn't divide b .

With the expression $|\{a_i\}|$ we denote the cardinality of the set of elements $\{a_i\}$.

In some passages the symbol $[x]$ indicates the greatest integer less then or equal to x , but in general the square brackets are used to group symbols in an expression.

4. Plan of the Work

The aim of the article is to calculate the number d such that the two integers $e \pm d$ were primes. We observe that if the number d is such that it satisfies the $\pi(\sqrt{2e})$ relations $d \not\equiv \pm e \pmod{p_i}$ this fact alone represents a sufficient condition for the desired result. Therefore the results provided in the following sections can be summarized by these two logical steps:

1. Given an even natural number $2e$ it will be provided a method in order to calculate an integer d such that $e \pm d \not\equiv 0 \pmod{p_i}$ where p_i are the $\pi(\sqrt{2e})$ primes less than or equal to $\sqrt{2e}$. This assure that the two integers $e \pm d$ are prime numbers. This step depends on the choice of a set of positive integers b_i and on the choice of their signs (+/-).
2. It will be provided a method in order to calculate at least one set of integer b_i such that $b_i \not\equiv \pm e \pmod{p_i}$, and considering that $d \equiv b_i \pmod{p_i}$ the consequence is that $d \not\equiv \pm e \pmod{p_i}$. This step, in combination with the step 1, allow us to obtain the number d greater than $-e$ and less than e . Consequently we obtain a pair of mirror primes respect to e , that is: $(e - d) + (e + d) = 2e$.

5. The Chinese Remainder Theorem and the Number d

About the existence of prime numbers in the intervals defined by the number e , that is: $(0, e)$ and $(e, 2e)$, the following result holds.

Lemma 1. *Given an arbitrary integer $e > 7$ there is at least one prime number in the intervals $(0, e)$ and $(e, 2e - 2)$.*

Proof. In the first interval we have trivially at least the first two primes. Regarding the second interval, actually it is the Bertrand's postulate, also called the Bertrand-Chebyshev theorem or Chebyshev's theorem: it states that if $e > 3$, there is always at least one prime p between e and $2e - 2$ [5]. \square

Furthermore we can ask if there are particular relations among $q_1, q_2, e, and d$. We can easily observe that the mirror primes can not be prime factors neither of the natural number $2e$ nor of d . We summarize those remarks in the following Lemma.

Lemma 2. *Given e , even integer number, for all pairs of mirror primes respect to e it holds: (a) $q_1 \nmid e$; (b) $q_2 \nmid 2e$; (c) $q_1 \nmid d$.*

Proof. We prove (a) by reductio ad absurdum. We have: $q_2 = 2e - q_1$, and if $q_1 \mid e$ then $q_1 \mid q_2$, but q_2 is prime and so it is impossible. The statement (b) follows from the identity $q_1 = 2e - q_2$ and from the fact that q_1 is by hypothesis prime. Similarly the third statement follows from observing that $q_2 = q_1 + 2d$. \square

Remembering that a prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself, it is easily to infer that every composit number n has a prime

factor less than or equal to \sqrt{n} . In fact if n has a prime factor less than or equal to its square root it is a composite number. Conversely let suppose for simplicity that n could be written, according to the fundamental theorem of arithmetic [11, 9], as the product of two primes a and b . If both of these two numbers were greater than \sqrt{n} we should have that: $n = a \cdot b > \sqrt{n} \cdot \sqrt{n} = n$, that is $n > n$, and it is not possible. The underlying rationale can be generalized to an arbitrary number of prime factors. So at least one prime divisors of each of the two quantity $e \pm d$, if composite, are in the interval $[1, \sqrt{e \pm d}]$. On the other hand if $e \pm d$ are not divisible by any prime less then or equal to $\sqrt{e \pm d} < \sqrt{2e}$ then they can not be a composite integer, so it has to be prime. The following lemma introduces the calculation method for the number d depending on an arbitrary choice of suitable integers b_i .

Lemma 3. *Given a natural number e and a set of $k = \pi(\sqrt{2e})$ arbitrary integers b_i , there exists an integer number d solution of the following system of congruencies:*

$$d \equiv b_i \pmod{p_i}, i = 1 \dots k$$

where p_i are the $k = \pi(\sqrt{2e})$ primes less then or equal to $\sqrt{2e}$.

Proof. Observing that $(p_i, p_j) = 1$ for $i \neq j$, the Chinese Remainder Theorem (CRT) [3] tell us that the system has a solution congruent to $p_1 \cdot p_2 \cdot \dots \cdot p_k$. The solution can be calculated by the following formula: $d = \sum_{1 \leq i \leq k} b_i P_i P'_i$, where: $P = p_1 \cdot p_2 \cdot \dots \cdot p_k$, $p_i = P/p_i$, and P'_i is the inverse of P_i , that is: $p_i P'_i \equiv 1 \pmod{p_i}$. \square

Let us note that the right terms of several congruencies could be considered as negative integers, in this case the result of the lemma is still valid (that is, supposing $b_i > 0$, we would have: $d \equiv -1 \cdot b_i \pmod{p_i}$ for some $i \in [1 \dots k]$, and the correspondent terms of the expression of the solution would be modified accordingly multiplying by -1). Supposing that we can choose a set of integer numbers b_i with $i = 1 \dots \pi(\sqrt{2e})$ such that $b_i \not\equiv \pm e \pmod{p_i}$, then the number d calculated by the foregoing method, if it is in the interval $[-(e-2), +(e-2)]$, is such that: $d \equiv b_i \pmod{p_i}$, and $b_i \not\equiv \pm e \pmod{p_i}$, $\Rightarrow d \not\equiv \pm e \pmod{p_i}$, that guarantees $e \pm d$ being a pair of primes.

6. How to Calculate the Integers b_i

The system of congruencies defined in the Lemma 3 provides an integer d that depends on the choice of the k integers b_i . Let us try to state the requirements they must meet. First of all we want the followings k conditions hold: $b_i \not\equiv \pm e \pmod{p_i}$, in order to assure that $e \pm d \not\equiv 0 \pmod{p_i}$, that is $p_i \nmid e \pm d$, which ensures that $e \pm d$ will be prime numbers. A possible choice for the number b_i could be that provided by the following lemma.

Lemma 4. *Let e be a positive integer, we denote p_1, p_2, \dots, p_k the first k primes with $k = \pi(\sqrt{2e})$, then the quantity defined as follows are not congruent to $\pm e$ modulo p_i for each i : $b_i = \left[\frac{e}{p_i} \right] p_i$, if $p_i \nmid e$, and $b_i = \left[\frac{e}{p_i^{\alpha_i}} \right] p_i + 1$, if $p_i \mid e$ with α_i the i -th prime power according to the fundamental theorem of arithmetic [11, 9], and $[x]$ indicates the greatest integer less then or equal to x .*

Proof. First consider the case in which $p_i \nmid e$. Observe that the expression $e - \frac{b_i}{p_i}$ represents the remainder after the division $\frac{e}{p_i}$, so it is not divisible by p_i . If $e \pm b_i$ was divisible by p_i we could write: $e = (h \mp \left\lfloor \frac{e}{p_i} \right\rfloor) p_i$, with h a generic integer, but this would mean that $p_i \mid e$, contra hypothesis. Now suppose $p_i \mid e$, similarly if $e \pm b_i$ was divisible by p_i we could write: $e = (h \mp \left\lfloor \frac{e}{p_i} \right\rfloor) p_i + 1$, with h a generic integer, but but this would mean that $p_i \nmid e$, contra hypothesis. \square

Unfortunately the choice of the numbers b_i as indicated by Lemma 4 doesn't assure that d is less than our integer e , let see for instance the following sidebar related to the integer 68.

$e = 68$

p_i	p_i	p'_i	b_i
2	1155	1	35
3	770	2	66
5	462	3	65
7	330	1	63
11	210	1	66

$d = 266805 \pmod{2310} = 1155$. $p_1 = d - e = 1087$, $p_2 = d + e = 1223$, both prime numbers.

Sidebar 1. Lemma 4 applied to the number $e = 68$

In order to avoid values of d greater than or equal to $e - 2$ and, because the solution d could be negative depending on the choice of the sign of b_i , less than or equal to $-(e - 2)$, the following constraints must hold:

$$Q_k \cdot B^T \leq e - 2$$

$$Q_k \cdot B^T \geq -(e - 2)$$

where

$$B_{1 \times k} = (\pm b_1 \pm b_2 \pm b_3 \dots \pm b_k),$$

$$Q_{k \times k} = \begin{bmatrix} p_1 \cdot P'_1 & 0 & 0 & \dots & 0 \\ p_1 \cdot P'_1 & p_2 \cdot P'_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ p_1 \cdot P'_1 & p_2 \cdot P'_2 & p_3 \cdot P'_3 & \dots & p_k \cdot P'_k \end{bmatrix},$$

with $\det(Q) \neq 0$.

Observe that the solution d of the system of congruencies of Lemma 3 is $Q_k \cdot B^T$, where Q_k is the k -th row of the matrix Q , where $k = \pi(\sqrt{2e})$. In order to calculate the k sets of integer b_i such that $b_i \not\equiv \pm e \pmod{p_i}$ we define the following sets of indexes i :

Definition 2. Let define the sets: $I_S = \{i : p_i \mid e\}$ and $I_R = \{i : p_i \nmid e\}$, where $1 \leq i \leq \pi(\sqrt{2e})$. Regarding the choice of b_i we have: If the index i is in I_S then we choose: $b_i \in \{1 \leq j \leq e - 2 : p_i \nmid (e \pm j)\}$, If the index i is in I_R then we chose: $b_i \in \{0\} \cup \{1 \leq j \leq e - 2 : p_i \nmid (e \pm j)\}$.

From these sets $\{b_j\}_i, i = 1 \dots k$, considering that for each positive element $+b_{ij}$ there is also the corresponding negative one $b_{ij+1} = -b_{ij}$, we can define the correspondent sets $\{w_j\}_i$ as $w_{ij} = b_{ij} \cdot P_i \cdot P'_i, i = 1 \dots k, j = 1 \dots |\{b_j\}_i|$.

Remark 1. Given an arbitrary integer $e > 7$, if there exists a choice of the quantities b_i , calculated according to Definition 2 (so that, if not equal to zero, they are positive and negative quantities), that satisfies the following relations:

$$Q_k \cdot B^T \leq e - 2 \quad (1)$$

$$Q_k \cdot B^T \geq -(e - 2) \quad (2)$$

where $d = Q_k \cdot B^T$ is the solution of the system of k congruencies: $d \equiv b_i \pmod{p_i}$, then the two integers $e - d$ and $e + d$ are prime numbers, that is there exists a pair of mirror primes respect to the integer e .

There is an obvious way to formulate the problem as a constraint satisfaction problem (CSP) [14, 16, 21], that is the research of a value, selected from a given finite domain, to be assigned to each variable so that all constraints relating the variables are satisfied. Defining $x_{ij} = 1$ if it is chosen the value w_{ij} we have:

$$d = \sum_{1 \leq i \leq k} \sum_{1 \leq j \leq |\{b_h\}_i|} w_{ij} \cdot x_{ij} \quad (3)$$

$$d \in [-(e - 2), (e - 2)] \quad (4)$$

$$\sum_{1 \leq j \leq |\{b_h\}_i|} x_{ij} = 1, \forall i \quad (5)$$

$$x_{ij} \in \{0, 1\} \quad (6)$$

The Definition 2 implies that if e is a prime number then all the sets $\{b_j\}_i$ have an element equal to zero, and the choice of d is trivially $d = 0$. The algorithm provided in the following check this zero-configuration as the first choice so it is able to determine the primality of the given number. The enormous number of combinations of the elements b_i that is possible to choose could be explored by a simple backtracking algorithm, and an example is provided by the pseudo-code in the sidebar 2.

Obviously this strategy is not an efficient option when the number e becomes large, in fact the search space becomes too large to search exhaustively. One way to reduce the combinations explored consists in gradually expanding the research domain, for example by one unit at each step for all the k sets, depending on a random choice. We can also ignore unfeasible nodes (combinations for which the number d is not in the feasible interval) observing that if d is greater than $e - 2$ when the current w is greater than zero we can jump to the next value of w (negative by definition). The algorithm modified in this way has been proven to be already more efficient for small number as reported in table 1. That is probably due to the fact that the first combination of number w_{ih} that satisfied both of the inequalities 1 and 2 is obtained for indexes $j \ll |\{b_j\}_i|$. This is just a conjecture, directly derived from the original GC, namely a stronger form of GC. Once verified by computer that the new conjecture is indeed satisfied

up to some enormous number, it may be easier to analyze this stronger form than the original GC. See Table 1 for a comparison of the two algorithms in terms of execution time for some small numbers (algorithms executed on an iOS platform with Intel Core 2 Duo SL9600). So this conjecture can be written as:

Statement 4. For every integer $e > 7$ if there exists one or more choices of w_{ij} such that $d = \sum_{1 \leq i \leq k} w_i$ is in the interval $[-(e-2), (e-2)]$, then for at least one choice we have that all the k numbers $|b_i|$ of the terms w_i ($|x|$ means here the absolute value of x) are less than or equal to ${}^{k/r}\sqrt{e}$, with r a real less than k .

This upper bound for the number of element of each set $\{b_j\}_i$ imply that the run-time complexity of the second algorithm is of $O(e^r)$, with $r < k$, much less than the case of the first one.

Remark 2. It could be tried to prove the conjecture by induction. If we consider the base case as $e = 8$, we have that $d = 3$ is obtained by the first and the second b_i , in particular as $b_{11} = 1$, $b_{22} = -3$, $d = 3$ and $r = 2/3$. If we suppose the conjecture true for e we have that for $e + 1$ holds for $i \in [1, k(e)]$:

$$P_i(e+1) = P_i(e), \text{ if } k(e) = k(e+1)$$

$$P_i(e+1) = P_i(e)P_{k(e+1)}, \text{ if } k(e) < k(e+1)$$

In the first case it is easy to demonstrate that for each indexes in the definition 2 related to the integer $e + 1$, it can be chosen either the same index as for e or one of the indexes $j - 1$ or $1 - j$, so that the number d is still in the desirable interval, and the fact that the k numbers $|b_i|$ for the number e are less than or equal to ${}^{k/r}\sqrt{e}$ implies that the k numbers $|b_i|$ for the number $e + 1$ are less than or equal to ${}^{k/r}\sqrt{e+1}$. The second case is more tricky, and we decided to attack it in a next research.

```

Initialize  $d$ , the vector  $c$  (pointer to the current elements of the set  $\{w_i\}_h, h = 1 \dots k$ )
while  $d \notin [-(e - 2), (e - 2)]$ 

     $h = k$ ;  $flag = 1$ ;
    while ( $flag == 1$ )

         $d = d - w_{h,c(h)}$ ;
         $c(h) = c(h) + 1$ ; /move forward
        if  $c(h) > |\{b_i\}_h|$ 

             $flag = 1$ ;
             $c(h) = 1$ ;
             $d = d + w_{h,c(h)}$ ;
             $h = h - 1$ ;

        else

             $flag = 0$ ;
             $d = d + w_{h,c(h)}$ ;

        end

    endwhile

endwhile

```

Sidebar 2. Algorithm based on a backtracking "depth-first" search

The core of the algorithms provided so far can be applied together with a simple heuristic as well. In particular we can focus the search in a neighborhood of the values w_i chosen such that they be of the same order of w_1 , for all the w_{1j} . Also the ordering of the set of w can influence the performance of the search, and we can order the rows w_i in descending order respecting to the mean of the elements of each row so that the first row contains the biggest values on average.


```

Initialize  $d$ , the vector  $c$  (pointer to the current elements of the set  $\{w_i\}_h, h = 1 \dots k$ ), the
vector  $depth$ 
while  $d \notin [-(e - 2), (e - 2)]$ 

     $depth_i = \min(depth_i + 2 * rand(1, k), |\{b_j\}_i|)$ ;
    while ( $h > 0$ )

         $h = k$ ;  $flag = 1$ ;
        while ( $flag == 1$ )

             $d = d - w_{h,c(h)}$ ;
             $c(h) = c(h) + 1$ ;
            while the node has been already explored
                 $c(h) = c(h) + 1$ ; /move forward
            endwhile
            if  $c(h) > depth(h)$ 
                 $flag = 1$ ;
                 $c(h) = 1$ ;
                 $d = d + w_{h,c(h)}$ ;
                 $h = h - 1$ ;
            else
                 $flag = 0$ ;
                if a unfeasible node has been reached then
                     $c(h) = c(h) + 1$ ; /move forward
                     $d = d + w_{h,c(h)}$ ;
                end
            endwhile
        endwhile
    endwhile
endwhile

```

Sidebar 3. Algorithm based on a forward checking approach

```

Initialize  $d$ , the vector  $c$  (pointer to the current elements of the set  $\{w_i\}_h, h = 1 \dots k$ ), the
vector  $depth$ 
while  $d \notin [-(e-2), (e-2)]$ 

     $c(1) = c(1) + 1$ ;
    /* Let's consider just the positive value of  $w_{h,c(h)}$ :
    while ( $wabs_{h,c(h)} < 0.98 * wabs_{1,c(1)}$ ) AND ( $c(h) < |\{b_j\}_h|$ )

         $c(h) = c(h) + 1$ ;

    endwhile
     $depth(h) = c(h) - 1$ ;
    while ( $h > 0$ )

         $h = k$ ;  $flag = 1$ ;
        while ( $flag == 1$ )

             $d = d - w_{h,c(h)}$ ;
             $c(h) = c(h) + 1$ ;
            if  $c(h) > depth(h) + 2$ 

                 $flag = 1$ ;
                 $c(h) = 1$ ;
                 $d = d + w_{h,c(h)}$ ;
                 $h = h - 1$ ;

            else

                 $flag = 0$ ;
                 $d = d + w_{h,c(h)}$ ;

            end

        endwhile

    endwhile

endwhile

```

Sidebar 4. Algorithm based on a simple heuristic that doesn't explore the entire tree

Number e	<i>Algorithm₁</i>		<i>Algorithm₂</i>	
	<i>t(sec)</i>	<i>d</i>	<i>t(sec)</i>	<i>d</i>
68	< 1	15	< 1	15
188	249	-105	132	-105
273	> 3600	-	646	206
368	> 3600	-	590	-231

Table 1: Comparison of the two algorithms for some small integers

Another way to view the problem is the following. In order to calculate the quantity d according to the Lemma 3, we have to calculate the set of b_i (positives by construction) as stated in Definition 2. Once obtained these positive integers, we have to attribute the appropriate sign to each one. The choice of the integer $+b_i$ versus $-b_i$ can be viewed as the choice of two sets of binary variable x_i and y_i , where:

$$x_i = \begin{cases} 0 & \Leftrightarrow y_i = 1 \Leftrightarrow \text{we choose } -b_i \\ 1 & \Leftrightarrow y_i = 0 \Leftrightarrow \text{we choose } +b_i \end{cases}$$

The constrains (1) can be written as:

$$\sum_{1 \leq i \leq k} w_i \cdot x_i + \sum_{1 \leq i \leq k} (-w_i) \cdot y_i \leq e - 2 \tag{7}$$

$$\sum_{1 \leq i \leq k} w_i \cdot x_i + \sum_{1 \leq i \leq k} (-w_i) \cdot y_i \geq -(e - 2) \tag{8}$$

where $w_i = b_i \cdot P_i \cdot P'_i, i = 1 \dots k$.

It may be observed that $y_i = 1 - x_i$, and then the preceding constrains, after substituting $y_i = 1 - x_i$, can be written as (in the following the symbol $[\cdot]$ represents just parentheses):

$$\sum_{1 \leq i \leq k} w_i \cdot x_i \leq \frac{1}{2} \left[(e - 2) + \sum_{1 \leq i \leq k} w_i \right] \tag{9}$$

$$\sum_{1 \leq i \leq k} w_i \cdot x_i \geq \frac{1}{2} \left[-(e - 2) + \sum_{1 \leq i \leq k} w_i \right] \tag{10}$$

Therefore it must be proved that, under the assumption that $e \geq 3$, the feasible set defined by the constraints is not empty. In other words, at least one choice of the binary variables x_i satisfies:

$$\begin{aligned} \sum_{1 \leq i \leq k} w_i \cdot x_i &\leq \frac{1}{2} \left[(e - 2) + \sum_{1 \leq i \leq k} w_i \right] = \frac{1}{2} \sum_{1 \leq i \leq k} w_i + E \\ \sum_{1 \leq i \leq k} w_i \cdot x_i &\geq \frac{1}{2} \left[-(e - 2) + \sum_{1 \leq i \leq k} w_i \right] = \frac{1}{2} \sum_{1 \leq i \leq k} w_i - E \end{aligned}$$

where $E = \frac{(e-2)}{2}$.

Now observe that the two constraints define a convex set. In fact, for all $\alpha \in (0, 1)$ and considering two elements of this set x_i^1 and x_i^2 satisfying the constraints, we have:

$$\sum_{1 \leq i \leq k} w_i \cdot (\alpha x_i^1 + (1 - \alpha)x_i^2) = \alpha \sum_{1 \leq i \leq k} w_i \cdot x_i^1 + (1 - \alpha) \sum_{1 \leq i \leq k} w_i \cdot x_i^2 \leq \frac{1}{2} \sum_{1 \leq i \leq k} w_i + E \tag{11}$$

$$\sum_{1 \leq i \leq k} w_i \cdot (\alpha x_i^1 + (1 - \alpha)x_i^2) = \alpha \sum_{1 \leq i \leq k} w_i \cdot x_i^1 + (1 - \alpha) \sum_{1 \leq i \leq k} w_i \cdot x_i^2 \geq \frac{1}{2} \sum_{1 \leq i \leq k} w_i - E \tag{12}$$

The problem to find a solution in this set is known in literature as the Convex Feasibility Problem [4, 15, 1]. It consists in finding a point in the intersection of convex sets. The common way for solving it is the relaxation-projection algorithm [1, 17].

However we are interested for our goal in proving the existence of a choice of the variables x_i . Let observe that we may order the terms w_i such that: $w_i \geq w_{i+1}, \forall i = 1 \dots k - 1$, and then we can determine the index h such that:

$$\sum_{1 \leq i \leq h-1} w_i < \frac{1}{2} \sum_{1 \leq i \leq k} w_i \tag{13}$$

$$\sum_{1 \leq i \leq h} w_i \geq \frac{1}{2} \sum_{1 \leq i \leq k} w_i \tag{14}$$

and therefore choose the variables as follows:

$$x_i = 1, i = 1 \dots h, x_i = 0, i = h + 1 \dots k$$

With this choice of the values of the k variables we may have (note that $E > 0$ and $k \geq 2$ if $e \geq 3$):

$$\sum_{1 \leq i \leq k} w_i \cdot x_i = \sum_{1 \leq i \leq h} w_i \geq \frac{1}{2} \sum_{1 \leq i \leq k} w_i > \frac{1}{2} \sum_{1 \leq i \leq k} w_i - E$$

And therefore the second constraint (11) is satisfied.

Regarding the first constraint (11) it holds if:

$$\frac{1}{2} \sum_{1 \leq i \leq k} w_i \geq \sum_{1 \leq i \leq h} w_i - E \tag{15}$$

Note that this inequality holds for the number $e = 16$ and the choice of the $w = [15, 30, 12]$ (see the sidebar 5) but not for the number $e = 68$ and a choice of the $w = [1155, 4620, 1386, 330, 210]$ (see the sidebar 1). Conversely it holds for the choice $w = [3465, 0, 1386, 990, 1050]$ (that is $d = 39$). On the other hand let observe that we may order the terms w_i such that: $w_i \leq w_{i+1}, \forall i = 1 \dots k - 1$, and then we can determine the index h such that:

$$\sum_{1 \leq i \leq h} w_i < \frac{1}{2} \sum_{1 \leq i \leq k} w_i \tag{16}$$

$$\sum_{1 \leq i \leq h+1} w_i \geq \frac{1}{2} \sum_{1 \leq i \leq k} w_i \tag{17}$$

and therefore choose the variables as follows:

$$x_i = 1, i = 1 \dots h, x_i = 0, i = h + 1 \dots k$$

With this choice of the values of the k variables we may have (note that $E > 0$ and $k \geq 2$ if $e \geq 3$):

$$\frac{1}{2} \sum_{1 \leq i \leq k} w_i > \sum_{1 \leq i \leq h} w_i = \sum_{1 \leq i \leq k} w_i \cdot x_i > \sum_{1 \leq i \leq k} w_i \cdot x_i - E$$

And therefore the first constraint (11) is satisfied. Regarding the second constraint (11) it holds if:

$$\frac{1}{2} \sum_{1 \leq i \leq k} w_i \leq \sum_{1 \leq i \leq h} w_i + E \quad (18)$$

In other terms, both these conditions now obtained (15 and 18) could define a particular heuristic in finding the appropriate set of w_i together with the particular choices of the sign (that is x_i , as in (13), or in (16)). We will examine such an algorithm in a next research.

Furthermore, instead of deriving a generic value for the quantity d , it could be required to identify the greatest or the smallest d with the properties we have discussed in the preceding. The natural way to do this is to write the related optimization problem in terms of integer linear program, and we can see that it is a special case of well known formulation classified in literature as "Knapsack Problem" [10]. In particular, once chose the numbers b_i :

$$\begin{aligned} & \text{maximize (or minimize) } W_{(1 \times k)} X_{(k \times 1)} \\ & \text{subject to: } WX \leq U \\ & \quad \quad \quad -WX \leq U' \\ & \quad \quad \quad x_i \in \{0, 1\} \end{aligned}$$

where:

$$\begin{aligned} U &= \frac{1}{2} \sum_{1 \leq i \leq k} w_i + E, \\ U' &= -\frac{1}{2} \sum_{1 \leq i \leq k} w_i + E, \\ W_{(1 \times k)} &= (w_1 \dots w_k), \\ X_{(k \times 1)} &= (x_1 \dots x_k)^T. \end{aligned}$$

Given a set of items, each with a weight and a value (in our case they are coincident and the problem is called "Subset sum problem"), determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most valuable items. Moreover, considering the two sets of constrains, we observe that various methods are known in literature in order to deal with negative weight [12]. Let see as an example the following sidebar where the calculus is provided for the integer 16.

$$e = 16$$

p_i	$I = I_S \cup I_R$	$\{b_j\}_{i=1\dots k}$
2	I_S	$\{\pm 1, \pm 3, \pm 5, \dots\}$
3	I_R	$\{0, \pm 3, \pm 6, \dots\}$
5	I_R	$\{0, \pm 2, \pm 3, \pm 5, \dots\}$

from the CRT:

p_i	P_i	P'_i
2	15	1
3	10	1
5	6	1

$$d_h = \pm b_1 \cdot 15 \pm b_2 \cdot 10 \pm b_3 \cdot 6; 1 \leq h \leq \prod_{1 \leq j \leq k} |\{b_j\}_i|$$

for example consider the following choices:

$$B = [1 \ 0 \ 0], d = 15 > e - 2$$

solution not feasible

$$B = [-1 \ 3 \ -2], d = 3 \leq e - 2$$

solution feasible

therefore if we can choose $d = 3$:

$$16 - 3 = 13, \text{ prime}$$

$$16 + 3 = 19, \text{ prime}$$

so that:

$$13 + 19 = 32 = 2 \cdot 16$$

Sidebar 5. How obtain the number d in the case of $e = 16$

7. Conclusion and future work

In the present article we have reconducted the problem of finding a pair of mirror primes respect to a given integer $e > 7$, at a well known problems of operations research. We have provided some algorithms to solve it as well. We derived a condition in terms of numbers w_i and e that can lead to a more efficient way to choose the appropriate set w_i . An interesting way of further research is both theoretical and practical. It will be of high interest the completion of the proof of statement 4 in order to have an upper bound in the number of elements in the search domain. Moreover it will be useful to have a detailed analysis of a more efficient algorithm based on a new heuristic, as stated in the last section, exploiting the conditions (15) and (18).

References

- [1] S. Agmon. *The relaxation method for linear inequalities*, *Canad. J. Math.* 6, 382–392, 1954.
- [2] M. Agrawal, N. Kayal, N. Saxena. *PRIMES is in P*, *Annals of Mathematics* 160, 2, 781–793, 2004.
- [3] T. M. Apostol, *Introduction to Analytic Number Theory*, Springer-Verlag, New York, 1976.
- [4] F. E. Browder. *Convergence theorems for sequences of nonlinear ooperators in Banach spaces*, *Mathematische Zeitschrift* 100, 3, 201–225, 1967.
- [5] P. Chebyshev. *Mémoire sur les nombres premiers*, *Mém. Acad. Sci. St. Pétersbourg* 7, 17–33, (1850) 1854. Reprinted as §1–7 in *Œuvres de P. L. Tschebychef*, Tome I. St. Pétersbourg, Russia: Commissionaires de l'Academie Impériale des Sciences, 51–64, 1899.
- [6] J. R. Chen and Y. Wang. *On the odd Goldbach problem*, *Acta Math. Sinica* 32, 702–718, 1989.
- [7] N. G. Chudakov. *On the Goldbach problem*, *Doklady Akademii Nauk SSSR* 17, 335–338, 1937.
- [8] T. Estermann. *On Goldbach's problem: proof that almost all even positive integers are sums of two primes*, *Proc. London Math. Soc.* 2, 44, 307–314, 1938.
- [9] Euclid. *Euclid's Element*, Book VII, propositions 30 and 32.
- [10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman. 1979.
- [11] C.F. Gauss and A. A. Clarke. (translator into English), *Disquisitiones Arithmeticae (Second, corrected edition)*, New York: Springer, 1986.

- [12] F. Glover. *A multiphase-Dual Algorithm for the Zero-One Integer Programming Problem*, Operations Research, 13, 16, 1965.
- [13] G. H. Hardy and J. E. Littlewood. *Some problems of partitio numerorum : III: on the expression of a number as a sum of primes*, Acta Math. 44, 1-70 (1923). Reprinted in Collected Papers of G. H. Hardy, Vol. I, 561-630, Clarendon Press, Oxford, 1966.
- [14] A.K. Mackworth. *Consistency in networks of relations*, Artificial Intelligence 8, 99–118, 1977.
- [15] St. Maruster, C. Popirlan. *On the regularity condition in a convex feasibility problem*, Nonlinear Analysis 70, 1923–1928, 2009.
- [16] U. Montanari. *Networks of constraints: fundamental properties and applications to picture processing*, Information Science 7, 95–132, 1974.
- [17] T.S. Motzkin and I.J. Schoenberg. *The relaxation method for linear inequalities*, Canad. J. Math. 6, 393–404, 1954.
- [18] T. Tao, *Every odd number greater than 1 is the sum of at most five primes*, eprint arXiv:1201.6656, 2012.
- [19] J. G. Van der Corput. *Sur l'hypothese de Goldbach*, Proc. Akad. Wet. Amsterdam 41, 76–80, 1938.
- [20] I. M. Vinogradov. *The Method of Trigonometrical Sums in the Theory of Numbers*, New York: Interscience, 1954.
- [21] D. Waltz. *Generating semantic descriptions from drawings of scenes with shadows* Technical Report AI271. MIT, MA. 1972.