Application of Combinatorics and Number Theory in Designing an Efficient Currency Denominations

Bimo Adityarahman Wiraputra 13517004[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13517004@std.stei.itb.ac.id

***Abstract*—A currency denomination system is a collection of numbers representable in a currency, whether in coin or banknotes. Change-making problem is a well-known computational problem in finding the minimum number of coins and/or banknotes needed to create a certain sums of money. While it is proved to be an NP-hard problem, most people will use a greedy approach in solving the change-making problem. This paper is interested in determining which currency denomination where the greedy approach of the change-making problem is optimal, presenting some notable mathematical results and algorithms regarding such denomination.**

***Keywords*— canonical coinage system, change-making problem, dynamic programming, greedy.**

I. Introduction

When we are buying or selling things in our routine, using real money, one or both parties involved will probably need to give the other party an exact amount of money using what amount of currency they have. The change-making problem is interested in the minimal number of coins and/or banknotes needed to create a certain sums of money given the denominations the currency has.

Formally, let the sequence $a_1 < a_2 < \cdots < a_n$ be the denomination available in the currency. In Rupiahs, for example, the available denomination will be $100, 200, 500, 1000, \dots$ Then given an amount W, we are interested in finding the sequence $x_1, x_2, \dots, x_n$ of nonnegative integers such that it achieves the right amount of sum:

$$\sum_{i=1}^{n} x_i a_i = W$$

while minimizing the value of cost, the number of coins/banknotes needed:

$$\sum_{i=1}^{n} x_i$$

In complexity theory, we know that the change-making problem is a special sub-problem of the famous Knapsack problem which is already proven to be NP-hard problem. Hence we know that the change-making problem cannot be solved in polynomial time with regard to n.

However when such issues are encountered in real life, most people will use the greedy approach in finding the value of $x_1, x_2, \dots, x_n$ such that the sum of the money will be equal to W. In such greedy approach, we try to diminish the value of W as fast as possible by always taking the highest value of available that is not greater than the remaining value of W until we reach the desired amount. For example, when we are trying to make a sum of 245,000 Rupiahs, we first take two copies of 100,000 banknotes, then we take two copies of 20,000, and lastly we take a copy of 5,000 rupiah and we are done here because we already have the desired sum.

While this is a fast algorithm in just $O(n)$ time complexity, we do not know for certain if such $x_1, x_2, \dots, x_n$ is optimal, that is a solution that achieves the minimal cost.

Now, we call a certain type of currency denomination system canonical if the greedy approach for all value of achievable W will yield the minimal number of coins/bank notes amount. We are then interested to find out the characteristics of a canonical system and the method as to how to determine if a certain system is canonical or not.

II. Basic Combinatorics and Number Theory Results

While there are several theorems presented in this paper, all of those theorems will not use any complicated mathematics results and will depend on these basic combinatorics and number theory results we all are familiar with.

A. Well-ordering principle

Well-ordering principle is a basic principle regarding the natural numbers, stating that every non-empty set of positive contains a least element. That is, such set contains an element that is the smallest, or smaller than the rest of the set's element. While sounding very simplistic, the well-ordering principle is equivalent to the mathematical induction.

To illustrate the equivalence, assume that the well-ordering principle holds. Suppose we have a proposition P where $P(1)$ is true, and $P(n)$ implies $P(n+1)$ for all positive integers n. If P does not holds true for all values of positive integers, then the set of positive integers n such that $P(n)$ is not true is non empty, therefore we have a smallest value a. Certainly $a \neq 1$, and we know that $a - 1$ is a positive integer not part of the set because it will contradicts the minimality of a. therefore we have $P(a-1)$ is true while $P(a)$ is not, contradicting $P(n) \rightarrow$

$P(n+1)$ for all positive integers n.

In the other way, if the mathematical induction holds true. Let S be a non empty set of positive integers and $P(n)$ be the proposition that there is no elements of S that is less than n. Certainly $P(1)$ holds, therefore if $P(n) \rightarrow P(n+1)$ for all n positive integers, by mathematical induction, we have $P(n)$ true for all positive integers n, which is ridiculous since S is non empty, hence there is a positive integer a in S, hence $P(a+1)$ cannot hold. Then there must be an x such that $P(x) \rightarrow P(x+1)$ statement is false, that is when $P(x)$ is true, and $P(x+1)$ is false. therefore there is no elements of S less than x, but there is that is less than $x+1$, which we can conclude that x is in S. Then we have a least element of S, that is x because all positive integers less than x is not in S.

Well-ordering principle or mathematical induction may or may not be called a theorem depending on the axiomatic framework of the natural numbers we are using. In Peano arithmetic, mathematical induction is an axiom while in axiomatic set theory, it is not taken as a fundamental axiom.

B. Linear Combination and Greatest Common Divisor

We define the greatest common divisor of two positive integers, a and b, commonly written as $\gcd(a,b)$, or just (a,b) if the context is clear, as the greatest number x such that x divides both a and b. Clearly x exists, as the number 1 certainly divides both a and b, and x cannot be bigger than a or b. If we consider both a and b by its unique prime factorization, then we can see that x is the number where in its prime factorization, the power of any prime is the minimum of the power of the prime in the factorization of a and b. From such consideration, we can also see the unique property of GCD, that is for all x dividing both, a and b, then we have that x must also divides (a,b).

Now consider the smallest positive integers that is the linear combination of a and b, that is the minimum positive value of $ax+by$ with x,y integers. We can prove that such minimum value is actually (a,b). To prove this, consider that (a,b) divides a and b, therefore it must also divides $ax+by$. Also, we can have the minimum value of $ax+by$ must also divides both a and b, because if not, say it does not divide a, then see that $a \; mod \; (ax+by)$ is a positive integers smaller than $ax+by$ and it is also a linear combination of a and b, which contradict the previous minimality. Therefore we have $ax+by$ divides (a,b). From which, we conclude the needed results.

III. Basic Currency Denominations Prerequisites

Before we delve into canonical system, we may consider the property a good currency denominations system has. We can consider the value representable in our currency system. Certainly given a denominations list $a_1 < a_2 < \cdots < a_n$, a value W is representable, or can be made, in our currency system if and only if it is a linear combination of $a_1, a_2, \ldots, a_n$ with all the coefficient $x_1, x_2, \ldots, x_n$ nonnegative integers. If we have the value W and V representable, certainly $W+V$ is also representable, because the nonnegative integers set is closed to addition. But, because the set is not closed under subtraction, we do not know for sure if $W-V$ is representable in our denominations list. Certainly it made sense to have the subtraction value also representable in context of change-making and other economical book-keeping practices, therefore we restrict our currency system such that the subtraction of two representable values must also be representable. Then we met upon our first theorem.

Theorem 1. In a currency denomination system $a_1 < a_2 < \cdots < a_n$, $W \geq V$ are representable implies $W-V$ representable, if and only if a_1 divides a_i for all i.

Proof.

First, if we have a_1 divides a_i for all i, then we certainly have

$$a_1 | \sum a_i x_i$$

for all x_i integers. So all representable value in the currency system must be a nonnegative multiple of a_1, then if $W \geq V$ are representable, then $W-V$ must also be a nonnegative multiple of a_1 which is obviously representable.

Next, because every integers can certainly be expressed as the difference between two nonnegative integers, we then have the consequence that all nonnegative integer W a linear combination of a_i,

$$W = \sum a_i x_i$$

for x_i integers must be representable in the system. Then for all i, because (a_1, a_i) is a nonnegative linear combination of a_1 and a_i by previous result, we then have the GCD representable as well, but because (a_1, a_i) is less or equal a_1, for it to be representable we must have $(a_1, a_i) = a_1$ for all i. Hence we have a_1 divides a_i for all i. ■

Because of theorem 1, we then have a_1 divides a_i for all i. Now see that W is representable in the system if and only if W is divisible by a_1. Therefore we can assume without loss of generality that $a_1 = 1$ by dividing everything in the context by the initial value of a_1. Hence we have all representable values as the set of nonnegative integers itself. From this point onward, we will assume that $a_1 = 1$ for any currency system (we care of).

IV. Naïve Calculation to Change-Making Problem

Given a currency system $1 = a_1 < a_2 < \cdots < a_n$ and a nonnegative integer W, we are interested in finding the total amount of coins/banknotes needed in a greedy solution, say $G(W)$ and the optimal solution, say $M(W)$.

Now we certainly have $G(0) = 0$, and by the nature of the greedy approach, we have for W a positive integer,

$$G(W) = G(W - a_i) + 1$$

where a_i is the greatest denomination not greater than W. Then from the formula, we can formulate a simple algorithm to count the value of $G(W)$ in $O(\text{n})$ time :

```
def G (a,W) :
  ans = 0
  i = len(a)
  while W > 0:
    ans += W/a[i]
    W %= a[i]
    --i
  return ans
```

The calculation of the $M(W)$ is a little more complicated considering it is not in polynomial time in regard of n from its NP-hardness, but from the property that $M(0) = 0$ and for W positive integer,

$$M(W) \leq M(W - a_i) + 1$$

for all i with equality happening if and only if there is an optimal representation of W using a_i. This formula is true, because given an optimal representation of $W - a_i$, adding one a_i denomination, we then have a representation of W. If there is an optimal representation of W using a_i, then removing that denomination, we have a representation of $W - a_i$ which must be optimal because if it is not, then substituting the representation of W with that optimal representation plus one a_i denomination creates a less costly representation of W, which is a contradiction. Hencefore we can use a dynamic programming using the result table for $M(W)$ to get a time complexity of (Wn) and memory complexity $O(W)$:

```
def O (a,W) :
  if res[W] >= 0 return res[W]
  // res[W] the lookup table we use that is
initialized with negative values
  if W == 0 return res[0] = 0

  ans = W
  // initialized ans with upper bound
answer
  for i in range(len(a)+1):
    if W-a[i] >= 0:
      ans = min(ans,O(W-a[i])+1)
  return res[W] = ans
```

Now, we call this complexity pseudo-polynomial time because its running time is polynomial to the size of the input, not the length of the input. That is why the use of canonical system will affect our runtime greatly because the big size of the input does not make the greedy algorithm runs any slower.

V. Canonical System

To give better illustration, we first look at examples of canonical systems and non-canonical systems. For canonical system, we have a simple example:

Theorem 2. A denomination system $1 = a_1 < a_2 < \cdots < a_n$ where a_i divides a_{i+1} for all i is canonical.

Proof.

Assume that it is not canonical, then by the well-ordered principle, there is a value W counterexample such that its greedy cost is greater than its optimal cost. Then see that the greedy representation of W must not contain a same denomination as an optimal representation of W. That is because removing that same denomination, we have a smaller value, say $W - a_i$ that has greedy cost the greedy cost of W minus one, and optimal cost the optimal cost of W minus one, so it is also a counterexample, contradicting the minimality of W. Now consider the smallest denomination used in its greedy representation, say a_i, and the smallest in its optimal representation, say a_j. Then they must be different. Assume first that $a_i < a_j$. Because a_i divides a_{i+1} for all i, we can easily prove by induction that a_i divides a_j. Using the same result, because $a_j \geq a_{i+1}$ the smallest denomination of W optimal representation, we then have a_{i+1} divides every denomination in that representation, so a_{i+1} divides W. Taking modulo a_{i+1}, we then have a_{i+1} divides $x_i a_i$ in W greedy representation. But that is impossible because if we use the greedy approach, we must then take a_{i+1}first until we do not take any a_i denomination. Contradiction. A similar contradiction will also appear for the case $a_i > a_j$ where we can then substitute $x_i a_i$ with fewer denominations of a_{i+1}. So the denomination system is indeed canonical. ■

Hence a denomination system like 1,2,6 or 1,5,10,50,100 is proven to be canonical. An example of a non-canonical system is 1,3,4 where the greedy representation of 6 is 1,4,4 with cost of 3 and the optimal representation is 3,3 with cost of 2.

Now, to determine if a certain system is canonical or not, considering we already have some algorithms for calculating the greedy cost and the optimal cost of a certain value from a given currency system, our intuition says that we just have to compare the two functions value for certain values of W. But before further results, we certainly cannot do that because the values of W we need to check go unbounded to infinity. Hence we need a bound for W that we need to check to prove if a certain system is canonical or not. A result from Dexter Kozen and Shmuel Zaks gives the answer to this problem, which we resume here:

Theorem 3. For a system $1 = a_1 < a_2 < \cdots < a_n$. If there exists an x such that $M(x) < G(x)$, then the smallest such x lies in the range

$$a_3 + 1 < x < a_{n-1} + a_n$$

Proof.

Certainly for $x < a_3$, the representation of x only consists of denomination a_1 and a_2, then because $a_1 = 1$ divides a_2, we know that by the previous result, the greedy cost will certainly be the same as its optimal cost. For $x = a_3$, both value must be 1 and for $x = a_3 + 1$, both value must be 1 (if $a_4 = a_3 + 1$) or 2 otherwise.

In the other case, for $x \geq a_{n-1} + a_n$, assuming all the other value under x satisfies $M(x) = G(x)$, take a_i any denomination used in an optimal representation of x. If $i = n$, we have

$$\begin{aligned} G(x) &= G(x - a_n) + 1 \\ &= M(x - a_n) + 1 \\ &= M(x) \end{aligned}$$

else, we have

$$\begin{aligned} G(x) &= G(x - a_n) + 1 && (G\ property) \\ &= M(x - a_n) + 1 && (assumption) \\ &\leq M(x - a_n - a_i) + 2 && (M\ property) \\ &= G(x - a_n - a_i) + 2 && (assumption) \end{aligned}$$

$$\begin{aligned} &= G(x-a_i)+1 && (G\ property) \\ &= M(x-a_i)+1 && (assumption) \\ &= M(x) && (M\ property) \\ &\leq G(x) && (M\ property) \end{aligned}$$

So we have $G(x) = M(x)$ for $x \geq a_{n-1} + a_n$. Hence from the two argument, we have the smallest counterexample must happen in between $a_3 + 1 < x < a_{n-1} + a_n$. ■

Furthermore, it is also explained that these result is tight as in there are infinite examples of systems where the smallest counterexample occurs at $a_3 + 2$ and at $a_{n-1} + a_n - 1$. Some examples are of the form $1, k, 2k - 2$ for $x = a_3 + 2$, and of the form $1, k, k + 1$ for $x = a_{n-1} + a_n - 1$ with $k \geq 3$.

After we have obtained the upper limit, then we just need to run the calculation algorithms to calculate the value of $M(W)$ and $G(W)$ for $1 \leq W < a_{n-1} + a_n$ and compare the result to prove if the currency system is canonical or not. We can also use some optimization considering that for the smallest counterexample, we have

$$\begin{aligned} &G(W) > M(W) \\ &G(W) > \min\big(M(W-a_i)\big) + 1 \\ &G(W) > \min\big(G(W-a_i)\big) + 1 \\ &G(W) > G\big(W-a_j\big) + 1 \end{aligned}$$

for some a_j not greater than W. So we obtain an algorithm with time complexity $O(na_n)$ and space complexity $O(a_n)$:

```
def Canon (a) :
  res[0] = 0
  n = len(a)
  for x in range a[n]+a[n-1]:
    i = n
    while a[i] > x:
      --i
    res[x] = res[x-a[i]]+1
    while i > 1:
      --i
      if res[x]>res[x-a[i]]+1:
        return false
  return true
```

Beside the approach above, we may also be interested in 'inducting' a canonical system. Suppose we already have a canonical system $1 = a_1 < a_2 < \cdots < a_n$, and we are interested if a system $1 = a_1 < a_2 < \cdots < a_n < a_{n+1}$ is canonical or not. While the above approach cannot be directly used to prove if a random system is canonical or not, because a canonical system might have a subsystem which is not canonical. For example the system 1,2,4,5,8 is canonical which is easily checked by the previous result, but the system 1,2,4,5 is not. Regarding the canonical system issue given above, it turns out to have a beautiful result that is found independently by different groups of people overtime:

Theorem 4 (One-point theorem). Suppose we have a canonical system $1 = a_1 < a_2 < \cdots < a_n$. For an integer $a_{n+1} > a_n$, the currency system $1 = a_1 < a_2 < \cdots < a_{n+1}$ is canonical if and only if $G(ma_n) \leq m$ with $m = \lceil a_{n+1}/a_n \rceil$ with regard to the new system.

Proof.

For clarity, let $G'(x) = M'(x)$ be the greedy and optimal cost with regard to the old system.

If the system is canonical, then certainly the greedy cost will be equal the optimal cost for every value. Now, assume that $G(ma_n) = M(ma_n)$. Certainly for value $x < a_{n+1}$, the representation will not use the denomination a_{n+1}, so because the rest of the system is canonical, we have $G(x) = M(x)$. Then we only need to divide it into two cases:

The first case is for $a_{n+1} \leq x < ma_n$. Because $a_n < a_{n+1}$, we have $ma_n = (m-1)a_n + a_n \leq a_m + a_n < 2a_m$. Then the representation of x will only use zero or one denomination a_{n+1}. Then we have,

$$\begin{aligned} G(x) &= G'(x - a_{n+1}) + 1 \\ M(x) &= \min(M'(x), M'(x - a_{n+1}) + 1) \\ &= \min(G'(x), G(x)) \end{aligned}$$

So we only need to prove $G'(x) \geq G(x)$. Now because we have $x \geq a_{n+1} > (m-1)a_n$, we have

$$G'(x) = G'(x - (m-1)a_n) + m - 1$$

and $G(x) = G'(x - a_{n+1}) + 1 = G'(x - a_{n+1} + a_n)$, then we have

$$\begin{aligned} &G'(x) - G(x) \\ &= G'(x - (m-1)a_n) - G'(x - a_{n+1} + a_n) + m - 1 \\ &\geq m - 1 - G'(ma_n - a_{n+1}) \\ &= m - G(ma_n) \geq 0 \end{aligned}$$

where the third line comes from the fact that $G' = M'$ follows the triangle inequality (two representation can be summed to create another representation, even though it might not be optimal) and the fourth line comes from the assumption.

Now for the second case where $x \geq ma_n$, it is sufficient to show that there exists an optimal representation of x where x_{n+1}, that is the coefficient of a_{n+1}, is non-zero. If we found such representation for all $x \geq ma_n$, we can then just induct down by removing one denomination of a_{n+1} until we reach the base case where $x < ma_n$ and concluding the theorem because we already proved such case.

From an optimal representation $x_1, x_2, \ldots, x_{n+1}$, we can repeatedly use the following transformation :

1. If $x_n \geq m$, we can replace those m denominations of a_n with its greedy representation. This way the number of total coins does not increase because $G(ma_n) = M(ma_n)$ and the value paid by denominators less than a_{n+1}decreases as we add the number of a_{n+1} denominator.
2. If we have the case that
$$\sum_{i=1}^{n-1} x_i a_i \geq a_n$$
we then change those denominations in the sum to its greedy representation. Once again the total coins does not increase and the value paid by denominators less than a_{n+1} does not increase, the number of a_n or a_{n+1} increases.

From these two transformation kinds, it is apparent that we cannot do infinite transformations. Then at one point we must reach an optimal representation where we cannot do any transformations, hence we have $x_n < m$ and $\sum_{i=1}^{n-1} x_i a_i < a_n$. From these two facts we can derive

$$\sum_{i=1}^{n} x_i a_i < a_n + (m-1)a_n = ma_n$$

Then because the total value x is no less than ma_n, we have the coefficient of a_{n+1} not zero. Hence proven. ■

Theorem 4 is a really powerful and useful result because using theorem 4, for example we can get a trivial proof of theorem 2, considering that if a_{n+1} is divisible by a_n, then we have $ma_n = a_{n+1}$, where obviously $G(ma_n) = 1 \leq m$.

We can also use theorem 4 to prove that the normal currency system adopted by Indonesia and many different countries all over the world like in the European Union and United States of America using the 1,2,5 scheme canonical, that is the system using denominations $1,2,5,10,20,50,100,\ldots$, (after dividing by the lowest denominator) by simple induction using theorem 4. When a_{n+1} is divisible by a_n, the proof as said is trivial, and when it is not, that is when $a_n = 2 * 10^k, a_{n+1} = 5 * 10^k$, it is easy to check that $G(ma_n) = G(6 * 10^k) = 2$ with representation $10^k, 5 * 10^k$. Then we can see that this justifies the greedy approach most people use when making changes in daily transactions.

The last approach this paper will show to you is a true polynomial algorithm to determine whether a certain currency system is canonical or not. It is presented in a paper by Pearson with time complexities of $O(n^3)$. This paper uses some differing paradigms than the other results we have on this paper.

Here given a currency system $a_1 > a_2 > \cdots > a_n = 1$ (take great notes that here the currency is listed decreasing, this has several advantage which will become apparent soon). Then a representation of a value W is represented using a vector $X = (x_1, x_2, \ldots, x_n)$, where we have $X \cdot A = W$ where $A = (a_1, a_2, \ldots, a_n)$, the currency vector. Then see that a greedy representation of W is a vector X such that $X \cdot A = W$ and being the lexicographically greatest between such vector (we may recall that comparing a vector X is greater than Y lexicographically if and only if there exist k such that $x_i = y_i$ for all $i < k$ and $x_k > y_k$). That is because when we always take the greatest denomination, we prioritize on the leftmost component of the vector representation.

Here we also define a representation vector $Y(W)$ an optimal representation where $Y \cdot A = W$ and the sum of its component is maximal, and it is also the lexicographically greatest between such vectors. Then we know that a $G(W) = M(W)$ if and only if $X(W) = Y(W)$.

Because adding the vector $(0,0,\ldots,1)$ to X make it lexicographically greater, ultimately we have that X operation preserves order. Now we define the notion $X \subseteq Y$ if every component of X is not greater than its corresponding component of Y. Then here we introduce a nice-looking lemma (which we actually already used before).

Lemma 1. Call U greedy if $U = X(U \cdot C)$ and optimal if $U = Y(U \cdot C)$. Then (a). if $U \subseteq V$ and V is greedy, then U is also greedy. (b). if $U \sqsubseteq V$ and V is optimal, then U is also optimal.

Proof

Note that vector addition preserves lexicographical order, that is $A \leq B \leftrightarrow A + C \leq B + C$. Now let U' be any representation of $U \cdot C$, then we have

$$\begin{aligned} U' \cdot C &= U \cdot C \\ (V - U + U') \cdot C &= V \cdot C, \quad \textit{by linearity} \\ V - U' + U' &\leq V, \quad \textit{since V is greedy} \\ U' &\leq U, \quad \textit{since addition preserves order} \end{aligned}$$

then since U is lexicographically greatest, it is greedy. (a) is proven.

Define $A \sqsubseteq B$ if $|A| > |B|$ or ($|A| = |B|$ and $A \leq B$). Then the optimal representation is the greatest under the $\sqsubseteq$ comparison. The comparison is also preserved under addition, so we can reuse the above proof by substituting the comparison. (b) is proven. ■

Now finally, consider for a currency system, the smallest counterexample of it being canonical, say w, where $G(w) > M(w)$ with a being the smallest of such value. The important result using lemma 1 is that its representation vector $X(w)$ and $Y(w)$ do not have the same components where its value is nonzero. That is the set of nonzero components of $X(w)$ and $Y(w)$ are disjoint. We can conclude this because if we they have, say i, where x_i and y_i are nonzero, then decrementing both vector its i component, we have two vector representation of the same value from which its greedy and optimum representations are different (derived from lemma 1). Then we have a smaller value than a that is also another counterexample. Hence we have a contradiction.

Now let i, j be the first and the last nonzero components of $Y(w)$. Then because $X(w) > Y(w)$, we then know that $X(w)$ has zero value on its i component, and there is a nonzero component in some earlier position. Now the following theorem characterizes $Y(w)$ greatly.

Theorem 5. $Y(w)$ has the same component values with $X(a_{i-1} - 1)$ in component 1 to $j - 1$, is one greater in component j. The remaining entries are all zero.

Proof.

First, see that because $X(w)$ has a nonzero component before the i component, we know that $w \geq a_{i-1}$. Then, see that if we decrement the y_j, then we obtain a representation of $w - a_j$ that is optimal, hence it is also greedy by minimality of w. Then because that representation is greedy, we then have $w - a_j < a_{i-1}$. Thus we get the following bounds:

$$w - a_j < a_{i-1} \leq w$$

Now suppose we have $X(a_{i-1} - 1) = (x_1, x_2, \ldots, x_n)$, then since $a_{i-1} - 1 \geq a_i$, then $x_i > 0$, thus if we decrement both x_i and y_i to get a greedy representation of $X(a_{i-1} - 1 - a_i)$ and $X(w - a_i)$. Then by the previous bounds, we have $X(a_{i-1} - 1 - a_i) < X(w - a_i)$. From henceforth we derive by the preservation of lexicographical order from vector addition,

$$X(a_{i-1} - 1) < Y(w).$$

Besides that, if we decrement y_j by one, we have a valid greedy representation of $X(w - a_j)$, from which we know from the first bound, $w - a_j \leq a_{i-1} - 1 \rightarrow X(w - a_j) \leq X(a_{i-1} - 1) \rightarrow X(w - a_j) \leq X(a_{i-1} - 1) < Y(w)$. See that $X(w - a_j)$ differs in only the j component, so if $X(a_{i-1} - 1)$ is between the two vectors, then it must not differs from them in first $j - 1$ components.

As we know from the start from the choosing of j, we know that $y_{j+1}, \ldots$ are all zero. Now because $X(a_{i-1} - 1) < Y(w)$, we have $x_k < y_k$ for some k. Because the first $i - 1$ components of x, y agree, and after component j, the component of y is zero, we must have $x_j < y_j$. But because $X(w - a_j) \leq X(a_{i-1} - 1)$, with similar argument, we have $y_j - 1 \leq x_j$. Then we conclude that $y_k = x_k + 1$. ■

Now as we have thoroughly characterizes the smallest counterexample of a currency system, in term of i, j. Then if we want to prove whether a system is canonical or not, we just need to consider the $O(n^2)$ amounts of possible smallest counterexample if there are any valid counterexamples or not. Because we already have the optimal representation, we just need to calculate the greedy representation which we know can

algorithm that works in $O(n^3)$ time complexity for validating a canonic currency system.

VI. Conclusion

From the paper, we know some algorithms to determine whether a currency denominations system is canonic or not. We also have verified that the current system used by Indonesia and many other countries in the world is canonical, therefore using a greedy approach will give you the optimal solution to the change-making problem.

VII. Acknowledgment

In this paper, the author thanks the Almighty God for His Grace and Guidance for me to be able to complete this paper. The author also thanks Mr. Rinaldi Munir as a lecturer of Discrete Mathematics IF2120. Besides that, the author also thanks his parents, his colleagues, and many other party related that has helped in the creation of this paper directly or indirectly.

References

[1] A. Niewiarowska, M. Adamaszek, Combinatorics of the Change-Making Problem. *European Journal of Combinatorics*, Vol. 31, Issue 1, Jan. 2010, pp 47-63

[2] M.J.Magazine, G.L.Nemhauser, L.E.Trotter Jr., When the Greedy Solution Solves a Class of Knapsack Problems, *Operations Research,* Vol. 23, No.2, (Mar.-Apr.,1975), pp.207-217

[3] D.Kozen, S.Zaks, Optimal Bounds for the Change-Making Problem, *Theoret. Comput.* Sci. 123 (1994), 377-388

[4] D.Pearson, A Polynomial-time Algorithm for the Change-Making Problem, Technical Report TR 94-1433, Department of Computer Science, Cornell University, June 1994

[5] L.J.Cowen, R.Cowen, A.Steinberg, Totally Greedy Coin Sets and Greedy Obstructions, *Electronic Journal of Combinatorics* 15 (2008), #R90

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Desember 2018

Bimo Adityarahman Wiraputra 13517004