# Optimal Bounds for the Change-Making Problem

Dexter Kozen[*]
Computer Science Department
Aarhus University
Denmark

Shmuel Zaks[†]
Computer Science Department
Technion
Haifa, Israel

November 21, 1991

**Abstract**

The *change-making problem* is the problem of representing a given value with the fewest coins possible.

We investigate the problem of determining whether the greedy algorithm produces an optimal representation of all amounts for a given set of coin denominations $1 = c_1 < c_2 < \cdots < c_m$. Chang and Gill [CG] show that if the greedy algorithm is not always optimal, then there exists a counterexample $x$ in the range

$$c_3 \ \le\ x\ <\ \frac{c_m(c_m c_{m-1} + c_m - 3c_{m-1})}{c_m - c_{m-1}}.$$

[*]Supported by the Danish Research Academy, the Nations Science Foundation, and the John Simon Guggenheim Foundation. On sabbatical from: Computer Science Department, Cornell University, Ithaca, New York 14853, USA.

[†]Supported by the ESPRIT II Basic Research Actions Program of the EC under contract No. 3075 (project ALCOM).

To test for the existence of such a counterexample, [CG] proposes computing and comparing the greedy and optimal representations of all $x$ in this range.

In this paper we show that if a counterexample exists, then the smallest one lies in the range

$$c_3 + 1 \; < \; x \; < \; c_m + c_{m-1},$$

and these bounds are tight. Moreover, we give a simple test for the existence of a counterexample that does not require the calculation of optimal representations.

In addition, we give a complete characterization of three-coin systems and an efficient algorithm for all systems with a fixed number of coins. Finally, we show that a related problem is *coNP*-complete.

## 1 Introduction

The *change-making problem* is the problem of representing a given value with the fewest coins possible from a given set of coin denominations. Unboundedly many coins of each denomination are available.

Formally, given a finite system $c_1 < c_2 < \cdots < c_m = n$ of positive integers (the *coins*) and a positive integer $x$, we wish to determine non-negative integer coefficients $x_i$, $1 \le i \le m$, so as to minimize

$$\sum_{i=1}^{m} x_i \tag{1}$$

subject to

$$x = \sum_{i=1}^{m} x_i c_i \; . \tag{2}$$

The sequence of coefficients $x_1, \ldots, x_m$ is called a *representation* of $x$. The quantity (1) that we wish to minimize is called the *size* of the representation. A representation is *optimal* if it is of minimum size. If $x_i > 0$, then we say that the coin $c_i$ is *used* in the representation. We restrict our attention here to systems containing a penny (*i.e.*, $c_1 = 1$), so that every $x$ has a representation.

The change-making problem is a form of knapsack problem. Martello and Toth devote an entire chapter to it in their text on knapsack problems [MT], and a good summary of the state of knowledge can be found there. In general, the problem is $NP$-complete when the coin values are large and represented in binary [L]; however, it can be solved in time polynomial in the number of coins and the value of the largest coin. In this regard, a number of algorithms have been investigated, the simplest of which is the *greedy algorithm*, which repeatedly takes the largest coin less than or equal to the amount remaining. Equivalently and more efficiently: for each of $i = m, m-1, \ldots, 2, 1$ in that order, let $x_i$ be the integer quotient $\lfloor x/c_i \rfloor$ and set $x := x \bmod c_i$. This produces the greedy representation in time $O(m \log n)$. Note that this is the unique representation $x_1, \ldots, x_m$ such that for all $i$, $1 < i \leq m$

$$\sum_{j=1}^{i-1} x_j c_j \; < \; c_i \; . \tag{3}$$

The greedy representation is not necessarily optimal. For example, given the system $1, 3, 4$, the greedy algorithm produces the representation $2, 0, 1$ for the number 6; this representation is of size 3, whereas the optimal representation is $0, 2, 0$ of size 2. For some systems, however, the greedy algorithm always produces an optimal representation for any given value; as a matter of practical interest, we note that this is the case for the system $1, 5, 10, 25, 50, 100$ of American coins and the system $1, 5, 10, 50, 100, 500$ of Israeli coins. The question thus arises: how does one determine whether the greedy algorithm is always optimal for a given system?

**Definition 1.1** Given a system of coins, let $o(x)$ denote the minimum size over all representations of the number $x$ in that system, and let $g(x)$ denote the size of the greedy representation of $x$. Following [MT], we call the system *canonical* if $g(x) = o(x)$ for all $x$. If a system is not canonical, then a value $x$ for which $o(x) < g(x)$ is called a *counterexample* for the system. $\square$

**Example 1.2** For any nonnegative integer $k$, the system $1, 2, 4, \ldots, 2^k$ is canonical. The *Fibonacci system* $1, 2, 3, 5, 8, \ldots, F_k$ is canonical, where $F^k$ is the $k^{\text{th}}$ Fibonacci number. The system $1, k, k+1$ for $k > 2$ is not canonical: the counterexample $2k$ has optimal representation $0, 2, 0$ of size 2, whereas the greedy representation is $k-1, 0, 1$ of size $k$.

Chang and Gill [CG] show that it suffices to search for a counterexample among the members of a certain finite set; if no counterexample is found in this set, then no counterexample exists and the system is canonical. The size of the set to be checked is polynomial in the largest coin value. Specifically,

**Theorem 1.3 (Chang and Gill [CG])** *Let $1 = c_1 < \cdots < c_m$ be any system of coins. If $o(x) = g(x)$ for all $x$ in the range*

$$c_3 \leq x < \frac{c_m(c_m c_{m-1} + c_m - 3c_{m-1}}{c_m - c_{m-1}} , \tag{4}$$

*then the system is canonical.*

In order to check for a counterexample in this set, Chang and Gill propose computing thy greedy and optimal representations of each element of the set and comparing their sizes. Martello and Toth comment [MT, p. 142]:

> The proof [of Theorem 1.3] is quite involved and will not be reported here. Furthermore, application of the theorem is very onerous, calling for optimality testing of a usually high number of greedy solutions.

**Example 1.4** Consider the system 1, 2, 4, 8, 10, 16 (this example is taken from [MT, Example 5.2, p. 143]). In order to test whether this system is canonical according to the algorithm of Chang and Gill, we must compute and compare the sizes of the greedy and optimal representations of all 385 values $x$ in the range (4). □

In Section 2 below we give two results that simplify the process of testing for the existence of a couterexample:

- We give tight bounds for Theorem 1.3. Specifically, we show that if a counterexample exists at all, then the smallest one lies in the range

  $$c_3 + 1 < x < c_m + c_{m-1} ,$$

  and these bounds are tight for an infinity of systems. Note that the upper bound is linear in the largest coin value, whereas (4) is cubic. Thus in order to cheek the system of Example 1.4, we need only check a set of size 20.

- We show that it is not necessary to compute optimal representations for the numbers in the given range as suggested by Chang and Gill.

There is a much simpler test involving only the sizes of the greedy representations, which are trivial to compute in time $O(n)$ using the recurrence

$$g(x) = 1 + g(x - c) \tag{5}$$

where $c$ is the largest coin value less than or equal to $x$.

These results give rise to an $O(mn)$ algorithm for testing whether a given system of coins is canonical.

In Section 3 we give a characterization of systems of three coins and a simple $O(\log n)$ test for determining when such a system in canonical.

In Section 4 we extend these results to systems with any fixed number of coins.

In Section 5 we consider the related problem of determining whether the greedy representation of a given number $x$ in a given system is optimal. We show that this problem is *coNP*-complete. It remains open whether there is an algorithm that is polynomial in $m$ and $\log n$ for testing whether a given system is canonical.

## 2 Optimal Bounds

In this section we derive optimal bounds for the change-making problem. Many of our arguments hinge on the following lemma, which describes the behavior of the function $o$.

**Lemma 2.1** *Let $1 = c_1 < \cdots < c_m$ be any system of coins. For all $x$ and coins $c_i \leq x$,*

$$o(x) \ \leq \ o(x - c_i) + 1 \ , \tag{6}$$

*with equality holding if and only if there exists an optimal representation of $x$ that uses the coin $c_i$.*

*Proof.* Certainly (6) holds, since any optimal representation of $x - c_i$ gives a representation of $x$ of size $o(x - c_i) + 1$ by adding one to the coefficient of $c_i$. If in addition $o(x) = o(x - c_i) + 1$, then the representation of $x$ so obtained is optimal and uses the coin $c_i$. Conversely, given an optimal representation of $x$ that uses $c_i$, we can obtain a representation of $x - c_i$ of size $o(x) - 1$ by subtracting one from the coefficient of $c_i$, and (6) implies that this representation is optimal. $\square$

**Theorem 2.2** *Let $1 = c_1 < \cdots < c_m$ be any system of coins. If there exists an $x$ such that $o(x) < g(x)$, then the smallest such $x$ lies in the range*

$$c_3 + 1 < x < c_m + c_{m-1} \ . \tag{7}$$

*Moreover, these bounds are tight.*

*Proof.* Certainly $o(x) = g(x)$ for all $x < c_3$, since $c_1, c_2$ is a canonical system. In addition, neither $c_3$ nor $c_3 + 1$ provides a counterexample, since in both cases the greedy representation is optimal. This establishes the lower bound.

To prove the upper bound, let $x \geq c_m + c_{m-1}$ and assume inductively that $g(y) = o(y)$ for all $y < x$. Let $c_i$ be any coin used in some optimal representation of $x$. If $i = m$, then

$$\begin{array}{rcll} g(x) & = & g(x - c_m) + 1 & \text{by definition of } g \\ & = & o(x - c_m) + 1 & \text{by induction hypothesis} \\ & = & o(x) & \text{by Lemma 2.1.} \end{array}$$

If $i < m$, then

$$\begin{array}{rcll} g(x) & = & g(x - c_m) + 1 & \text{by definition of } g \\ & = & o(x - c_m) + 1 & \text{by induction hypothesis} \\ & \leq & o(x - c_m - c_i) + 2 & \text{by Lemma 2.1} \\ & \leq & g(x - c_m - c_i) + 2 & \text{by definition of } o \\ & = & g(x - c_i) + 1 & \text{by definition of } g \\ & = & o(x - c_i) + 1 & \text{by induction hypothesis} \\ & = & o(x) & \text{by Lemma 2.1} \\ & \leq & g(x) & \text{by definition of } o. \end{array}$$

Thus in either case $g(x) = o(x)$.

For $k > 2$, the systems 1, $k$, $2k - 2$ give an infinity of systems for which the smallest counterexample is $c_3 + 2$, and the systems 1, $k$, $k + 1$ give an infinity of systems for which the smallest counterexample is $c_m + c_{m-1} - 1$. Thus the bounds (7) are tight. $\square$

Our simplified algorithm is based on the observation that we can avoid computing optimal representations by checking for the existence of *witnesses* instead of counterexamples:

**Definition 2.3** A *witness* is an $x$ for which

$$g(x) > g(x-c)+1$$

for some coin $c < x$. □

**Lemma 2.4** *(i) Every witness is a counterexample.*

*(ii) If a counterexample exists, then the smallest one is a witness.*

*Proof.*

(i) Suppose $x$ is a witness; thus

$$g(x-c)+1 < g(x)$$

for some coin $c$. Then

$$\begin{array}{rcll} o(x) & \leq & o(x-c)+1 & \text{by Lemma 2.1.} \\ & \leq & g(x-c)+1 & \text{by definition of } o \\ & < & g(x) & \end{array}$$

(ii) If $x$ is a counterexample but not a witness, and if $c$ is any coin used in an optimal representation of $x$, then $x-c$ is also a counterexample:

$$\begin{array}{rcll} o(x-c) & = & o(x)-1 & \text{by Lemma 2.1.} \\ & < & g(x)-1 & \\ & \leq & g(x-c) & \end{array}$$

Therefore the smallest counterexample must be a witness.

□

The converse of Lemma 2.4(i) is false: in the system $1,4,5$ the value 12 is a counterexample but not a witness. In this example, the coin 4 is used in the optimal representation $0,3,0$ of 12, therefore $8 = 12-4$ is also a counterexample. It is in fact the smallest counterexample, thus is also a witness.

**Theorem 2.5** *For a given system to be canonical, it is necessary and sufficient that there exist no witness in the range (7).*

*Proof.* Immediate from Theorem 2.2 and Lemma 2.4. □

Theorem 2.5 implies that to test whether a given system is canonical, it suffices to check whether

$$g(x) \ \leq \ g(x-c)+1$$

for all $x$ in the range (7) and coins $c < x$; we need not calculate any optimal representations. All necessary values of $g(x)$ can be computed in time $O(n)$ using the recurrence (5); thus the entire algorithm takes time $O(mn)$.

## 3 A Characterization of Three-Coin Systems

In this section we characterize completely all systems of three coins. This characterization gives a trivial $O(\log\, n)$ test for determining whether the system is canonical.

Let $1 < c < d$ and let $q$ and $r$ be the quotient and remainder, respectively, obtained from the integer division of $d$ by $c$. Thus $q$ and $r$ are the unique integers such that

$$\begin{aligned} d &= qc+r\ , \qquad (8)\\ 0 \ \leq \ r &< \ c\ . \qquad (9) \end{aligned}$$

**Theorem 3.1** *The system* $1, c, d$ *is not canonical if and only if* $0 < r < c-q$.

*Proof.* If $0 < r < c-q$, then the value $d+c-1$ is a counterexample: the greedy representation $c-1, 0, 1$ is of size $c > r+q$, whereas the representation $r-1, q+1, 0$ is of size $r+q$.

Conversely, suppose $1 < c < d$ is not canonical, and let $x$ be the smallest counterexample. The greedy representation of $x$ must be of the form $e, 0, 1$ with $0 < e < c$, since $d+1 < x < c+d$ by Theorem 2.2. Moreover, there is a unique optimal representation of $x$ of the form $0, k, 0$ with $k > 0$, since if either the coefficient of 1 or $d$ were nonzero, then by Lemma 2.1 we could subtract one and get a smaller counterexample. Since $x = d + e = kc$, we have

$$\begin{gathered} d = kc - e = (k-1)c + (c-e) \\ 0 \ < \ (d+c) - x = (d+c) - (d+e) = c - e \ < \ c \ , \end{gathered}$$

and since $q$ and $r$ are unique numbers satisfying (8) and (9), we must have $q = k-1$ and $r = c-e$. Since $x$ is a counterexample, we have that $k < 1+e$, thus $q = k-1 < e$ and $0 < c-e = r$, from which the desired inequalities $0 < r < c-q$ follow. □

## 4 Large Coins

The characterization of the previous section yields a simple $O(\log n)$ algorithm for determining whether a given system of three coins is canonical. In this section we give an algorithm whose time complexity is $O(\log n)$ for any fixed number of coins $m$. The complexity of the algorithm is $O(m^2 2^{m-1} \log n)$.

Recall that $\lfloor x/c \rfloor$ and $x \bmod c$ denote the integer quotient and remainder, respectively, obtained when dividing $x$ by $c$. Thus

$$\begin{aligned} x &= \lfloor x/c \rfloor c + x \bmod c \\ 0 &\leq x \bmod c < c \end{aligned}$$

and $\lfloor x/c \rfloor$ and $x \bmod c$ are the unique numbers for which these two statements hold.

Let $\gamma_i(x)$ denote the greedy representation of $x$ in the system $1 = c_1 < \cdots < c_i$. Thus

$$\begin{aligned} \gamma_1(x) &= x \\ \gamma_i(x) &= \langle \gamma_{i-1}(x \bmod c_i), \lfloor x/c_i \rfloor \rangle, \quad i > 1 \end{aligned}$$

where $\langle \alpha, z \rangle$ denotes the sequence obtained by appending the integer $z$ to the end of the sequence $\alpha$.

Define the equivalence relation $\equiv_k^i$ on integers $x \geq k$ by:

$$x \equiv_k^i y \ \leftrightarrow \ \gamma_i(x) - \gamma_i(x-k) = \gamma_i(y) - \gamma_i(y-k) \ ,$$

where $-$ applied to the sequences $\gamma_i(\ )$ denotes componentwise difference. Note that $x \equiv_{c_m}^m y$ for every $x, y \geq c_m$. It follows from the observation

$$g(x) - g(x-c) \ = \ \sum_{i=1}^{m} (\gamma_m(x) - \gamma_m(x-c))_i$$

that if $x \equiv_c^m y$ for a coin $c$, then $x$ satisfies the property

$$g(x) \quad \leq \quad g(x-c)+1 \tag{10}$$

if and only if $y$ does. Thus in order to find a witness, it suffices to check (10) for one representative $x$ from each $\equiv_c^m$-class for each coin $c$. We will show below (Theorem 4.2) that for each coin $c$ there are at most $2^{m-1}$ $\equiv_c^m$-classes, and representatives can be constructed efficiently.

The formal statement and proof of Theorem 4.2 do not adequately reflect the intuition behind them, so we preface the formalities with the following intuitive argument.

Fix $k$ and consider the difference $\gamma_m(x)-\gamma_m(x-k)$ of the greedy representations of $x$ and $x-k$ as $x$ increases. The last coefficient of this difference, namely $\lfloor x/c_m \rfloor - \lfloor (x-k)/c_m \rfloor$, alternates periodically between two values $r$ and $r+1$ (unless $k$ is a multiple of $c_m$, in which case there is only one value). We can thus think of $x$ as being in one of two states, depending on the value of this coefficient. The state changes whenever either $x$ or $x-k$ skips over a multiple of $c_m$. In between the times when this state changes, the next-to-last coefficient of $\gamma_m(x)-\gamma_m(x-k)$ alternates periodically between two states in a similar fashion, but with period $c_{m-1}$; and so on. Thus each coin value $c_i$, $i \geq 2$, accounts for two states (there is only one state for $c_1 = 1$), giving $2^{m-1}$ global states.

Formally, let $x, y$, and $c$ be integers, $c$ positive. Define

$$t_c(x,y) = \lfloor (x \bmod c + y \bmod c)/c \rfloor \ \in \ \{0,1\} \ .$$

The function $t_c$ formalizes the "state" for coin $c$ as described above. The following lemma establishes some basic properties of this function.

**Lemma 4.1** *The function $t_c$ satisfies the following properties:*

$$\begin{aligned}
(x+y) \bmod c \quad &= \quad x \bmod c + y \bmod c - c\, t_c(x,y) && (11)\\
\lfloor (x+y)/c \rfloor \quad &= \quad \lfloor x/c \rfloor + \lfloor y/c \rfloor + t_c(x,y) && (12)\\
t_c(x,y) = 0 \quad &\leftrightarrow \quad x \bmod c \leq (x+y) \bmod c && (13)\\
t_c(x,y) = 1 \quad &\rightarrow \quad t_c(y+x,-x) = 0 \ . && (14)
\end{aligned}$$

*Proof.* For (11),

$$\begin{aligned}&x \bmod c + y \bmod c\\ &= \lfloor (x \bmod c + y \bmod c)/c \rfloor c + (x \bmod c + y \bmod c) \bmod c\\ &= c\, t_c(x,y) + (x+y) \bmod c\ .\end{aligned}$$

For (12),

$$\begin{aligned}&\lfloor (x+y)/c \rfloor c + (x+y) \bmod c\\ &= x+y\\ &= \lfloor x/c \rfloor c + x \bmod c + \lfloor y/c \rfloor c + y \bmod c\\ &= \lfloor x/c \rfloor c + \lfloor y/c \rfloor c + ct_c(x,y) + (x+y) \bmod c\end{aligned}$$

by (11), thus

$$\lfloor (x+y)/c \rfloor = \lfloor x/c \rfloor + \lfloor y/c \rfloor + t_c(x,y)\ .$$

For (13), if $t_c(x,y) = 0$ then

$$\begin{aligned}x \bmod c &= (x+y) \bmod c - y \bmod c\\ &\le (x+y) \bmod c\ ,\end{aligned}$$

and if $t_c(x,y) = 1$ then

$$\begin{aligned}x \bmod c &= (x+y) \bmod c + c - y \bmod c\\ &= (x+y) \bmod c + (-y) \bmod c\\ &> (x+y) \bmod c\ ,\end{aligned}$$

since $(-y) \bmod c > 0$ (otherwise $y \bmod c = 0$, which would imply that $t_c(x,y) = 0$). Finally, for (14), we have by (13) that

$$\begin{aligned}t_c(x,y) = 1 &\rightarrow y \bmod c > (x+y) \bmod c\\ &\rightarrow (x+y) \bmod c < ((x+y) + (-x)) \bmod c\\ &\rightarrow t_c(x+y, -x) = 0\ .\end{aligned}$$

□

Define the sets

$$\begin{aligned}A_k^1 &= \{k\}\\ A_k^i &= \{\lfloor k/c_i \rfloor c_i + u \mid u \in A_{k \bmod c_i}^{i-1} \cup \{k + v \mid v \in A_{(-k) \bmod c_i}^{i-1}\},\ i > 1\ .\end{aligned}$$

**Theorem 4.2** *The set $A^i_k$ contains the minimum element of each $\equiv^i_k$-class. In other words, for all $x \geq k$ there exists a $y \in A^i_k$ such that*

$$k \leq \quad y \quad \leq x \tag{15}$$

$$y \quad \equiv^i_k \quad x\ . \tag{16}$$

*Proof.* The proof is by induction on $i$. The basis is immediate from the definition of $A^1_k$ and $\gamma_1$.

For $i > 1$, let $t_i = t_{c_i}$. We break the proof into two cases, depending on the value of $t_i(k, x-k)$. First suppose $t_i(k, x-k) = 0$. Then $k \bmod c_i \leq x \bmod c_i$. By the induction hypothesis, there exists a $u \in A^{i-1}_{k \bmod c_i}$ such that

$$k \bmod c_i \leq u \leq x \bmod c_i \tag{17}$$

$$u \equiv^{i-1}_{k \bmod c_i} x \bmod c_i\ . \tag{18}$$

Let

$$y = \lfloor k/c_i \rfloor c_i + u \in A^i_k\ .$$

By (17) and the fact that $k \leq x$, we have

$$\begin{aligned} k &= \lfloor k/c_i \rfloor c_i + k \bmod c_i \\ &\leq \lfloor k/c_i \rfloor c_i + u \quad (= y) \\ &\leq \lfloor x/c_i \rfloor c_i + x \bmod c_i \\ &= x. \end{aligned}$$

This establishes (15). By Lemma 4.1, we also have that $t_i(k, y-k) = 0$, since

$$k \bmod c_i \leq u = y \bmod c_i\ .$$

By (18) and the fact that $t_i(k, x-k) = t_i(k, y-k) = 0$, we have

$$\begin{aligned} &\gamma_{i-1}(x \bmod c_i) - \gamma_{i-1}((x-k) \bmod c_i) \\ &\quad = \gamma_{i-1}(x \bmod c_i) - \gamma_{i-1}(x \bmod c_i - k \bmod c_i) \\ &\quad = \gamma_{i-1}(u) - \gamma_{i-1}(u - k \bmod c_i) \\ &\quad = \gamma_{i-1}(y \bmod c_i) - \gamma_{i-1}(y \bmod c_i - k \bmod c_i) \\ &\quad = \gamma_{i-1}(y \bmod c_i) - \gamma_{i-1}((y-k) \bmod c_i)\ . \end{aligned} \tag{19}$$

Now suppose $t_i(k, x-k) = 1$. By Lemma 4.1, $t_i(x, -k) = 0$, thus $(-k) \bmod c_i \leq (x-k) \bmod c_i$. By the induction hypothesis, there exists a $v \in A^{i-1}_{(-k) \bmod c_i}$ such that

$$(-k) \bmod c_i \leq \; v \; \leq (x-k) \bmod c_i \tag{20}$$

$$v \equiv^{i-1}_{(-k) \bmod c_i} (x-k) \bmod c_i \; . \tag{21}$$

Let

$$y = k + v \in A^i_k.$$

By (20) and the fact that $k \leq x$, we have

$$\begin{aligned} k \;&\leq\; k + v \;\; (= y) \\ &\leq\; k + (x-k) \bmod c_i \\ &\leq\; x \; . \end{aligned}$$

This establishes (15). We also have that $t_i(k, y-k) = 1$:

$$\begin{aligned} k \bmod c_i + (y-k) \bmod c_i \;&=\; k \bmod c_i + v \bmod c_i \\ &\geq\; k \bmod c_i + (-k) \bmod c_i \\ &=\; c_i \; , \end{aligned}$$

since $k \bmod c_i \neq 0$ by Lemma 4.1(13). By (21) and the fact that $t_i(k, x-k) = t_i(k, y-k) = 1$, we have

$$\begin{aligned} &\gamma_{i-1}(x \bmod c_i) - \gamma_{i-1}((x-k) \bmod c_i) \\ &\quad= \; -(\gamma_{i-1}((x-k) \bmod c_i) - \gamma_{i-1}(x \bmod c_i)) \\ &\quad= \; -(\gamma_{i-1}(v) - \gamma_{i-1}(v - (-k) \bmod c_i) \\ &\quad= \; \gamma_{i-1}(y \bmod c_i) - \gamma_{i-1}((y-k) \bmod c_i) \; . \end{aligned} \tag{22}$$

Now for either value of $t_i(k, x-k)$, we have $t_i(k, x-k) = t_i(k, y-k)$. Then by Lemma 4.1,

$$\begin{aligned} \lfloor x/c_i \rfloor - \lfloor (x-k)/c_i \rfloor \;&=\; \lfloor k/c_i \rfloor + t_i(k, x-k) \\ &=\; \lfloor k/c_i \rfloor + t_i(k, y-k) \\ &=\; \lfloor y/c_i \rfloor - \lfloor (y-k)/c_i \rfloor \; . \end{aligned} \tag{23}$$

Thus in either case, using (19), (22), and (23), we have

$$\begin{aligned}
&\gamma_i(x) - \gamma_i(x-k) \\
&\quad = \langle \gamma_{i-1}(x \bmod c_i), \lfloor x/c_i \rfloor \rangle - \langle \gamma_{i-1}((x-k) \bmod c_i), \lfloor (x-k)/c_i \rfloor \rangle \\
&\quad = \langle \gamma_{i-1}(x \bmod c_i) - \gamma_{i-1}((x-k) \bmod c_i), \lfloor x/c_i \rfloor - \lfloor (x-k)/c_i \rfloor \rangle \\
&\quad = \langle \gamma_{i-1}(y \bmod c_i) - \gamma_{i-1}((y-k) \bmod c_i), \lfloor y/c_i \rfloor - \lfloor (y-k)/c_i \rfloor \rangle \\
&\quad = \langle \gamma_{i-1}(y \bmod c_i), \lfloor y/c_i \rfloor \rangle - \langle \gamma_{i-1}((y-k) \bmod c_i), \lfloor (y-k)/c_i \rfloor \rangle \\
&\quad = \gamma_i(y) - \gamma_i(y-k) \ ,
\end{aligned}$$

which establishes (16). $\square$

It is easily shown by induction that the set $A_k^m$ contains at most $2^{m-1}$ elements, and each element of $A_k^m$ is less than

$$\sum_{i=1}^{m} c_i \leq k + mn.$$

Moreover, the straightforward method of constructing $A_k^m$ according to its inductive definition takes time $O(m2^{m-1} \log n)$. Thus to check whether the system is canonical, we need only determine (10) for all coins $c$ and $x \in A_c^m$. There are $m2^{m-1}$ such $x$ to check, and each check takes time $O(m \log n)$.

## 5 An $NP$-Completeness Result

Lueker [L] shows that when the coin values are large and represented in binary, the problem of finding an optimal representation of a given $x$ is $NP$-hard. Here we show:

**Theorem 5.1** *It is coNP-complete to determine, given a system of coins and a number $x$ represented in binary, whether the greedy representation of $x$ is optimal.*

*Proof.* The problem is clearly in *coNP*: we can compute the greedy representation of $x$ in liner time, then find a better one if it exists by guessing.

To show *coNP*-hardness, we will encode the problem of *exact cover by three-sets*: given a set $X$ and a family $\mathcal{E}$ of three-element subsets of $X$, can

$X$ be represented as a disjoint union of elements of $\mathcal{E}$? This problem is known to be *NP*-complete (see [GJ]).

Assume without loss of generality that $X = \{1, 2, \ldots, 3n\}$. Let $p = n + 1$. Consider the system of coins

$$\begin{aligned} c_A &= 1 + \sum_{i \in A} p^i \ , \quad A \in \mathcal{E} \\ c_X &= \sum_{i=1}^{3n} p^i \end{aligned}$$

and a penny. Let

$$x = n + c_X \ .$$

The greedy algorithm gives a representation of $x$ of size $n + 1$ consisting of $c_X$ and $n$ pennies. This is optimal unless there is an exact cover, in which case a better representation is obtained by taking $c_A$ for $A$ in the cover. $\square$

The problem of Theorem 5.1 differs from the problem of determining whether a given system of coins is canonical in that in the former, we are asking whether greedy is optimal for a given $x$, whereas in the latter, we are asking whether greedy is optimal for all $x$. We know by Theorems 2.5 and 5.1 that both problems are in *coNP*, and the former is complete. The burning question that we have not succeeded in answering is whether the latter is complete, or whether there is an algorithm whose time complexity is polynomial in $m$ and $\log n$.

## 6 Acknowledgements

We thank Gudmund Frandsen, Kim Skak Larsen, Peter Bro Miltersen, Erik Meineche Schmidt, and Sven Skyum for their contributions.

## References

[CG] S. K. Chan and A. Gill, "Algorithmic solution of the change-making problem," *j. Assoc. Comput. Mach.* 17:1 (January 1970), 113–122.

[GJ] M. R. Garey and D. S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[L] G. S. Lueker, "Two *NP*-complete problems in nonnegative integer programming," Report No 178, Computer Science Laboratory, Princeton University, 1975.

[MT] S. Martello and P. Toth. *Knapsack problems.* John Wiley and Sons, 1990.