# Can our coinage system be improved?

PETER SHIU

*In loving memory of Christine Marian Shiu*

Some forty years ago my wife Christine and I considered the problems in this article, which involves a fair amount of computation. Computing facilities were not good then, so we considered instead the problems in [1] in which we showed, without using computers, that there were 64703 ways to make up £1 using coins; this was before the introduction of the 20p and £1 coins, and the $\frac{1}{2}$p coin was in circulation. If Christine were still with us, this would have been another piece of joint work. I therefore dedicate this article to her memory.

## 1. *Introduction*

The design of a coinage system depends on considerations we give to various criteria: for example, the number of denominations for the coins, the maximum number of coins required to deliver any given amount in a range, or the required number of coins averaged over the range; see also §3.

Consider our current system of 1p, 2p, 5p, 10p, 20p and 50p, besides the £1 and £2 coins, which will not concern us here. For $0 \leqslant n \leqslant 99$, denote by $r(n)$ the least number of coins required to deliver $n$ pence. It can be verified that $r(n) \leqslant 6$, with equality if, and only if, $n = 88, 89, 98, 99$, and that

$$r(0) + r(1) + \dots + r(99) = 340,$$

which shows that the average number of coins needed to deliver $n$ pence is 3.4. Is there a system, still with six coins of different values, which requires fewer than six coins to represent every $n$, and perhaps fewer than three coins on average? The answer is: Yes!

It was reported that the previous Chancellor (George Osborne) wanted to have the 1p coin abolished, but the suggestion was vetoed by the then Prime Minister (David Cameron); if the proposal had been accepted, then the amounts 1p and 3p could not be delivered, of course. In a general setting, however, the mathematical problems associated with similar proposals can become seriously difficult.

I started preparing this article after receiving the invitation from the British Congress of Mathematics Education (BCME 9) to give a presentation, and I wrote programs in *Python* for the required computations. It then occurred to me that there might be published articles on similar problems, and indeed there are. Thus, much of what I consider here is in the interesting article [2] by Jeffrey Shallit.

## 2. *Notation*

We assume that the range concerned is $0 \leqslant n \leqslant 99$, and that there is a coin with unit value, so that $n = 1$ can be represented. We denote a system with $k$ coins having the values $c_1, c_2, \dots, c_k$ by the vector

$$C_k = (c_1, c_2, \dots, c_k), \qquad 1 = c_1 < c_2 < \dots < c_k < 100,$$

and we restrict ourselves to $k = 4, 5, 6$. For example, the Japanese, Albanian and British systems are

$$C_4 = (1, 5, 10, 50), \; C_5 = (1, 5, 10, 20, 50), \; C_6 = (1, 2, 5, 10, 20, 50),$$

respectively. A choice of coins to represent $n$ can then be denoted by the vector

$$D_k = (d_1, d_2, \dots, d_k), \qquad d_i \geqslant 0,$$

if the scalar product $C_k \cdot D_k$ is $n$, that is

$$C_k \cdot D_k = c_1 d_1 + c_2 d_2 + \dots + c_k d_k = n;$$

the number $r$ of coins used is then the sum of the components of $D_k$, that is

$$r = r(n, D_k) = d_1 + d_2 + \dots + d_k.$$

The assumption that $c_1 = 1$ ensures that there is always at least one admissible choice, namely $(n, 0, 0, \dots, 0)$, with the largest $r$ equal to $n$.

For a given system $C_k$ our task is to determine, for each $n$ with $0 \leqslant n \leqslant 99$, the least corresponding $r$ among all choices $D_k$ that deliver $n$, that is the number

$$r(n) = \min_{C_k \cdot D_k \equiv n} r(n, D_k).$$

It would be unreasonable to expect an explicit formula for $r(n)$, but there should be a (simple) procedure to find the choice $D_k$ that delivers $r(n)$. We now set

$$A = A(C_k) = \sum_{0 \leqslant n \leqslant 99} r(n),$$

so that $A/100$ is the average number of coins used to represent $n$ in the range.

Finally, for a fixed $k$, the *optimal coinage problem* is to find the system $C_k$ which minimises the sum $A$, that is

$$A_k = \min_{C_k} A(C_k).$$

## 3. *The greedy algorithm*

For a general system $C_k$, with $k$ no longer small and a long range for $n$, the task of finding the optimal choice $D_k$ for $r(n)$ can be formidable. To bypass the difficulty, one approach is to make the simple, and easy to apply, choice by always taking the largest number of the largest denomination coin to reduce the current value before making use of the next smaller value coin.

Thus, for such a choice

$$D_k' = (d_1', d_2', \dots, d_k'), \qquad \text{with} \qquad r'(n) = d_1' + d_2' + \dots + d_k',$$

we first set

$$d_k' = \left\lfloor \frac{n}{c_k} \right\rfloor,$$

where $\lfloor z \rfloor$, is the greatest integer not exceeding the fraction $z$, and then set

$$d'_{h-1} = \left\lfloor \frac{n - d_k c_k - d_{k-1} c_{k-1} - \dots - d_h c_h}{c_{h-1}} \right\rfloor, \; h = k, k - 1, \dots, 2.$$

Take, for example, $n = 93$ in the British system. Applying the procedure we first take 50p, leaving 43p, so we take two 20p pieces next, leaving 3p, for which we use a 2p coin and then a 1p. Thus $D_6' = (1, 1, 0, 0, 2, 1)$, $C_6 \cdot D_6' = 1 + 2 + 0 + 0 + 40 + 50 = 93$ and $r'(93) = 5$. We call this the *greedy algorithm* to arrive at $r'(n)$. It turns out that, for this system, the greedy algorithm always delivers the optimal choice, that is $r'(n) = r(n)$ for $0 \leqslant n \leqslant 99$.

As we know from life, being greedy in a situation may not be the correct strategy to arrive at a desired optimal solution. Thus, there are systems $C_k$ for which the greedy algorithm fails to deliver the optimal choices. A simple counterexample is $C_3 = (1, 4, 9)$ with $n = 12$; the greedy algorithm then gives $D_3' = (3, 0, 1)$, with $r'(12) = 4$, whereas the optimal choice is $D_3 = (0, 3, 0)$ with $r(12) = 3$. It may be argued that the applicability of the greedy algorithm to deliver the optimal solution is also important for choosing a coinage system. Anyway, using $'$ to indicate that the minimisation process is restricted to applying the greedy algorithm, we now define

$$A_k' = \min_{C_k} A'(C_k), \qquad \text{where } A'(C_k) = \sum_{0 \leqslant n \leqslant 99} r'(n).$$

4. *A variety of systems*

The decimal notation being almost universal, most countries have adopted systems $C_k = (1, c_2, \dots, c_k)$, with $k = 4, 5, 6$, and $c_i \in \{2, 4, 5, 10, 20, 25, 50\}$, the proper divisors of 100, and $c_i = 10$ for some $i$. There are only four systems for $k = 6$ if exactly one member from each of the pairs $\{4, 5\}$ and $\{20, 25\}$ is taken. For such systems, the greedy algorithm always delivers the optimal solution, that is $r'(n) = r(n)$.

The Azerbaijani system is $C_6 = (1, 3, 5, 10, 20, 50)$, and one may question the wisdom of taking $c_2 = 3$, which does not divide 100. Actually, divisibility has little to do with the operation of the system, because the problem is the partitioning of numbers in the range, so that addition is of paramount importance. Indeed, the Azerbaijani system is essentially the same as the British system in that $A' = A = 340$ holds for both systems; the two values for $r(n)$ are slightly different for some $n$, of course.

We list most of the systems $C_k = (1, c_2, \dots, c_k)$ mentioned here in Table 1, including the Azerbaijani system. The number $J' = J'(C_k)$ in the fourth column is the largest number of coins required, and the last column gives the numbers $n$ for which $J'$ coins are required. Although $C_6 = (1, 2, 4, 10, 25, 50)$ has the least $A' = 336$, and with only three $n$ requiring six coins, I am not aware that it has been adopted by any country.

| $k$ | $C_k$ | $A'$ | $J'$ | $n$ |
|---|---|---|---|---|
| 4 | (1, 5, 10, 50) | 500 | 10 | 99 |
| 4 | (1, 5, 10, 25) | 470 | 9 | 94, 99 |
| 5 | (1, 5, 10, 20, 50) | 420 | 8 | 89, 99 |
| 5 | (1, 5, 10, 25, 50) | 420 | 8 | 94, 99 |
| 6 | (1, 2, 4, 10, 20, 50) | 340 | 6 | 87, 89, 97, 99 |
| 6 | (1, 2, 4, 10, 25, 50) | 336 | 6 | 92, 94, 98 |
| 6 | (1, 2, 5, 10, 20, 50) | 340 | 6 | 88, 89, 98, 99 |
| 6 | (1, 2, 5, 10, 25, 50) | 340 | 6 | 93, 94, 98, 99 |
| 6 | (1, 3, 5, 10, 20, 50) | 340 | 6 | 87, 89, 97, 99 |

TABLE 1

| $k$ | $C_k$ | $A'$ | $J'$ | $n$ |
|---|---|---|---|---|
| 4 | (1, 3, 11, 37)* | 410 | 7 | 67, 69, 93, 95 |
| 4 | (1, 3, 11, 38)* | 410 | 7 | 68, 70, 95, 97 |
| 5 | (1, 3, 7, 16, 40) | 346 | 6 | 68, 77, 92 |
| 5 | (1, 3, 7, 16, 41) | 346 | 6 | 69, 78, 94 |
| 5 | (1, 3, 7, 18, 44)* | 346 | 6 | 74, 78, 85 |
| 5 | (1, 3, 7, 18, 45) | 346 | 6 | 75, 79, 86 |
| 5 | (1, 3, 8, 20, 44)* | 346 | 6 | 77, 79, 82 |
| 5 | (1, 3, 8, 20, 45) | 346 | 6 | 78, 80, 83 |
| 6 | (1, 2, 5, 11, 25, 62) | 313 | 5 | 44, 45, 58, 59, 81, 82, 95, 96 |
| 6 | (1, 2, 5, 11, 25, 63) | 313 | 5 | 44, 45, 58, 59, 82, 83, 96, 97 |
| 6 | (1, 2, 5, 13, 29, 64) | 313 | 5 | 50, 51, 53, 54, 85, 86, 88, 89 |
| 6 | (1, 2, 5, 13, 29, 65) | 313 | 5 | 50, 51, 53, 54, 86, 87, 89, 90 |

TABLE 2

In Table 2, we remove the restriction that $c_i$ has to divide 100, apply the greedy algorithm to each system $C_k$, listing those with the least $A'$. Thus, $A' = 313$ and $J' = 5$ are achieved by four systems $C_6$. An asterisk denotes a system for which the greedy algorithm delivers the optimal solution with $r'(n) = r(n)$.

Are there systems for which $A < 300$? The answer is 'yes', there are in fact many, and even one with $J = 4$; we explain how they can be found in the next section.

## 5. *Computing the optimal choice*

Let $C_k$ be a system, and $0 \leqslant n \leqslant 99$. The following is the procedure for finding $r(n)$. We have $r(0) = 0$, $r(c_i) = 1$ for 1, 2, … , $k$, of course, and, for convenience, we also set $r(\ell) = 100$ if $\ell < 0$. Suppose that $r(m)$ have been computed for all $m < n$. Then $r(n)$ can be computed by the following reduction formula:

$$r(n) = \min_{1 \leqslant i \leqslant k} \{r(n - c_i)\} + 1.$$

To see that the formula is valid, let us first consider a target $n > c_k$. If the minimum here is attained at $i = j$, then the formula delivers $r(n) = r(n - c_j) + 1$, and the representation takes the form $n = (n - c_j) + c_j$; in other words, making use of the known optimal representation for $n - c_j$, we need only insert one more coin, namely $c_j$, to deliver $n$. This value for $r(n)$ is correct because a smaller value would imply that $r(n - c_j)$ was not optimal. The case when $n \leqslant c_k$ is the same, except that we may ignore the terms $r(n - c_i)$ if $n < c_i$.

For $k = 4$, 5, 6, we apply the optimisation process to each system $C_k$, giving the results for the least $A$ in Table 3. There are 7429 systems $C_6$ with $A < 300$, including $C_6 = (1, 4, 6, 21, 30, 37)$ with $J = 4$; the 28 numbers $n$ requiring four coins are:

$$\begin{cases} 15, 17, 19, 20, 24, 50, 53, 54, 56, 65, 69, 70, 76, 77, 82, \\ 83, 84, 85, 86, 87, 89, 91, 92, 93, 94, 96, 98, 99. \end{cases}$$

| $k$ | $C_k$ | $A$ | $J$ | $n$ |
|---|---|---|---|---|
| 4 | (1, 5, 18, 25) | 389 | 6 | 14, 82, 83, 88, 89, 92, 96, 99 |
| 4 | (1, 5, 18, 29) | 389 | 6 | 14, 27, 74, 79, 85, 91, 96, 98 |
| 5 | (1, 5, 16, 23, 33) | 329 | 6 | 14 |
| 6 | (1, 4, 6, 21, 30, 37) | 292 | 4 | (the 28 numbers above) |
| 6 | (1, 5, 8, 20, 31, 33) | 292 | 5 | 88 |

TABLE 3

6. *The Frobenius Problem*

The general representation problem becomes fraught with difficulty when there is no coin of unit value. Consider first the following related question: What is the largest number of chicken nuggets that cannot be purchased if such nuggets are sold only in packs of 6, 9, 20? The answer is 43.

*Proof*: Since $6 \equiv 9 \equiv 0 \pmod 3$, and $20 \equiv 2 \pmod 3$, we need at least two packs of 20 in order to deliver any $m \equiv 1 \pmod 3$, from which it follows that 43 is not representable as a non-negative linear combination of 6, 9, 20. Moreover, it can be readily verified that the next six consecutive numbers $m = 44, 45, 46, 47, 48, 49$ are representable as such. Now we are done because if $m \geqslant 50$ then there is a suitable $u \geqslant 0$ such that $m - 6u$ is one of these six numbers.

In a lecture in 1880, Frobenius posed the following:

*Frobenius Problem*: Let $a_1, a_2, \dots a_n > 1$, with $\gcd(a_1, a_2, \dots a_n) = 1$. Find a formula for, or a method to compute, the largest integer $M$ not representable as

$$a_1x_1 + a_2x_2 + \dots + a_nx_n, \qquad x_1, x_2, \dots .x_n \geqslant 0.$$

Note that if $\gcd(a_1, a_2, \dots, a_n) = d > 1$ then numbers not divisible by $d$ cannot be represented. In 1882 J. J. Sylvester gave the solution $M = ab - a - b$ to the case $n = 2$; we use $(a, b)$ and $(x, y)$ instead of $(a_1, a_2)$ and $(x_1, x_2)$ when $n = 2$.

*Proof* (Sketch): Suppose, if possible, that $M = ab - a - b$ is representable as $ax + by$. Then, since $a$, $b$ are coprime, $y \equiv -1 \pmod a$ and $x \equiv -1 \pmod b$, so that

$$M = ax + by \geqslant a(b - 1) + b(a - 1) = 2ab - a - b = M + ab.$$

This is impossible, so $M$ is not representable. A bit more work on the same idea shows that every $m > M$ is representable; the detail, and some related problems, are given in [3].

The special case $n = 3$ and $(a, b, c) = (6, 9, 20)$ is the chicken nuggets problem, which has the solution $M = 43$. Formulae for $M$ that cover all cases of $a$, $b$, $c$ have very recently been given by A. Tripathi [4]. For a fixed $n \geqslant 4$, R. Kannan [5] has given a method to determine $M$, but the implementation of the process is complicated, even for $n = 4$. Moreover, it has been shown by J. L. Ramírez-Alfonsín [6] that, for a general $n$, the Frobenius Problem is 'NP-hard under Turing reduction'; we do not explain such terms here, except to say that the hardness is similar, or equivalent, to that for the notorious *Travelling Salesman Problem.*

*References*

1. C. M. Shiu and P. Shiu, Stamps and coins: two partition problems, *Mathematical Spectrum* **13** (1980) pp. 49-55.
2. Jeffrey Shallit, What this country needs is an 18 cents piece'. *The Mathematical Intelligencer*, (2) **25** (2003) pp. 20-23.
3. P. Shiu, Moment sums associated with binary quadratic forms, *Amer. Math. Monthly* **113** (2006) pp. 545-550.
4. A. Tripathi, Formulae for the Frobenius number in three variables, *J. Number Theory* **170** (2017) pp. 368-389.
5. R. Kannan, Lattice translates of a polytope and the Frobenius problem, *Combinatorica* **12** (2) (1992) pp. 161-177.
6. J. L. Ramírez-Alfonsín, *The Diophantine Frobenius problem*, Oxford University Press (2006).

10.1017/mag.2018.53

PETER SHIU
*353 Fulwood Road, Sheffield S10 3BQ*
e-mail: *p.shiu@yahoo.co.uk*