A cryptarithmetic is something like

\[
\begin{array}{c}
\text{SEND} \\
\text{MORE} \\
\hline
\text{MONEY}
\end{array}
\]

For the letters, you must substitute distinct digits such that the words become numbers. In the above example, \text{MONEY} must be the sum of \text{SEND} and \text{MORE}.

We only consider cryptarithms of the above type, e.g. with two summands.

D. Eppstein already showed in [1] that Cryptarithms is NP-complete. He reduced from 3-SAT, but did not show ASP-completeness. Furthermore, his proof relies on a classical combinatorial result. We reduce from (1 in 3)-SAT instead. See [2] for the meaning of ASP-completeness.

Let us first assume that the letters do not need to be distinct digits. Then we have a variant of cryptarithms which is ASP-complete for any base $\geq 2$. The following symbolic digits on the right ensure that $c_i = i$ for all given $c_i$, where $b$ is the base:

\[
\begin{array}{c c c c c}
\vdots & c_0 & c_0 & c_1 & c_0 \\
\vdots & c_0 & c_0 & c_{b-1} & c_0 \\
\vdots & c_0 & c_1 & c_0 & c_0
\end{array}
\]

We write 0, 1, $b-1$ instead of $c_0, c_1, c_{b-1}$ from now. Let $a_1, a_2, \ldots, a_n \in \{0,1\}$ be the variables of our instance of 3-SAT and $a'_i := 1 - a_i$ the inverse of $a_i$. The following symbolic digits enforce that $a_i \in \{0,1\}$ and $a'_i = 1 - a_i$:

\[
\begin{array}{c c c c c}
\vdots & 0 & a_i & 0 & \cdots \\
\vdots & 0 & a'_i & 0 & \cdots \\
\vdots & 0 & 1 & 0 & \cdots
\end{array}
\]
Say that the \( j \)th equation denotes \( a'_2 + a_4 + a_7 = 1 \) (all others are similar). This equation can be coded as

\[
\begin{array}{cccccccc}
\cdots & 0 & t_j & 0 & a'_2 & 0 & \cdots \\
\cdots & 0 & a_7 & 0 & a_4 & 0 & \cdots \\
\cdots & 0 & 1 & 0 & t_j & 0 & \cdots \\
\end{array}
\]

So we can reduce (1 in 3)-SAT to cryptarithms without distinct digits for any base \( \geq 2 \).

Next, we show that Cryptarithms with distinct digits is ASP-complete, again by reducing from (1 in 3)-SAT. Since there are only \( b! \) possibilities to check, the base \( b \) can not be bounded. Again, write \( a_1, a_2, \ldots, a_n \) for the variables of an arbitrary instance of (1 in 3)-SAT and \( a'_1, a'_2, \ldots, a'_n \) for their inverses.

Take the base \( b \) at least \( 8(n^2 + 3n + 1) \) such that \( 4 \mid b \). We will use the following symbolic digits:

- \( c_0 = 0, \ c_4 = 4, \ c_8 = 8, \ldots, \ c_{b-4} = b - 4 \) and \( c_1 = 1 \) and \( c_2 = 2 \),
- \( \bar{a}_i = a_i + 4i + 1, \ \bar{a}'_i = a'_i + 4i + 1, \ \hat{a}_i = a_i + 4(n + 1)i + 1 \) and \( \hat{a}'_i = a'_i + 4(n + 1)i + 1 \), for all \( i \) with \( 1 \leq i \leq n \),
- \( d_i = 4(n(n + 2) + i) + 2 + a_i b/2 \) and \( d'_i = 4(n(n + 2) + i) + 2 + a'_i b/2 \) for all \( i \) with \( 1 \leq i \leq n \),
- \( t_{i,j} = \min(\bar{a}_i + \hat{a}_j, \bar{a}'_i + \hat{a}'_j) \) and \( t_{i,j} = \min(\bar{a}_i + \hat{a}_j, \bar{a}'_i + \hat{a}'_j) \) for some \( i \neq j \) with \( 1 \leq i, j \leq n \).

If the rightmost part of the cryptarithm looks like

\[
\begin{array}{cccccccc}
\cdots & c_0 & c_0 & c_{b-4} & c_0 & c_{b-8} & c_0 & \cdots \\
\cdots & c_0 & c_0 & c_4 & c_0 & c_0 & \cdots \\
\cdots & c_0 & c_1 & c_0 & c_0 & \cdots \\
\end{array}
\]

then the base \( b \) and all \( c_i \) are determined. Again, we write \( i \) instead of \( c_i \) from now. Since the \( a_i \) are no digits in our cryptarithm, we may define \( a_i := \bar{a}'_i \mod 2 \). The following enforces \( \bar{a}_i, \bar{a}'_i, \hat{a}_i, \hat{a}'_i, d_i \) and \( d'_i \) to have their
given values:

\[
\begin{array}{cccccccc}
\cdots & 0 & \bar{a}_i & 0 & \bar{a}_i & \bar{d}_i & 0 & \bar{a}_i & \cdots \\
\cdots & 0 & 2 & \bar{a}_i & \bar{d}_i & \bar{d}_i' & 0 & \bar{a}_i & \cdots \\
\cdots & 0 & \bar{d}_i & 0 & \bar{d}_i & \bar{d}_i' & 0 & \bar{a}_i & \cdots \\
\end{array}
\]

Since \( \bar{a}_i \equiv a_i \pmod{2} \), \( d_i + \bar{d}_i \) relieves a carry if and only if \( a_i = 1 \) and \( d_i \) has the desired value. Next, \( d_i' \) is what it should be, so \( \bar{a}_i \) also. At last, \( \bar{a}_i \) and \( \bar{a}_i' \) are determined.

Assume that \( a'_k + a_l + a_m = 1 \) is an equation of the given instance of (1 in 3)-SAT (all other equations are similar). We code this equation as follows:

\[
\begin{array}{cccccccc}
\cdots & 0 & t_{k,l}' & 0 & \bar{a}_k' & 0 & \cdots \\
\cdots & 0 & \bar{a}_m & 0 & \hat{a}_l & 0 & \cdots \\
\cdots & 0 & \bar{d}_k' & 0 & t_{k,l}' & 0 & \cdots \\
\end{array}
\]

This enforces the equation \( a'_k + a_l + a_m = 1 \) to be satisfied. Next, assume that \( a_k + a'_l + a_p = 1 \) is another such equation. If we code this equation, we use the same variable \( t_{k,l}' \) as above, which might seem odd. But, since

\[
2 = (a_k + a'_k) + (a_l + a'_l) \leq (a'_k + a_l + a_m) + (a_k + a'_l + a_p) = 2
\]

it follows that \( a'_k + a_l = a_k + a'_l = 1 \) and \( a_m = a_p = 0 \). Consequently, \( \bar{a}_k' + \hat{a}_l = \bar{a}_k + \bar{a}_l' = t_{k,l}' \).

So we only need to show that no collisions of symbolic digits occur, i.e. all symbolic digits are distinct. Notice first that every sum of at most one
\( \bar{a}_i \) and at most one \( \hat{a}_j \) is different, since \((i, j) \mapsto 4i + 4(n+1)j\) is injective if \(0 \leq i, j \leq n\). So all \( \bar{a}_i, \bar{a}'_i, \hat{a}_j, \hat{a}'_j, t_{i,j}, t'_{i,j} \) are different. Furthermore, the \( d_i \) and \( d'_i \) are larger than \( 4(n+1)^2 \), so they are larger than all previous symbolic digits. \( \min(d_n, d'_n) = 4n(n + 3) + 2 \) must not relieve a carry when added to itself, so \( b > 8n(n+3) + 4 \) i.e. \( b \geq 8(n^2 + 3n + 1) \).

References
