Fuzzy Simultaneous Congruences

Max A. Deppert  Kiel University  Kiel, Germany  made@informatik.uni-kiel.de
Klaus Jansen  Kiel University  Kiel, Germany  kj@informatik.uni-kiel.de
Kim-Manuel Klein  Kiel University  Kiel, Germany  kmk@informatik.uni-kiel.de

February 19, 2020

Abstract

We introduce a very natural generalization of the well-known problem of simultaneous congruences. Instead of searching for a positive integer $s$ that is specified by $n$ fixed remainders modulo integer divisors $a_1, \ldots, a_n$ we consider remainder intervals $R_1, \ldots, R_n$ such that $s$ is feasible if and only if $s$ is congruent to $r_i$ modulo $a_i$ for some remainder $r_i$ in interval $R_i$ for all $i$.

This problem is a special case of a 2-stage integer program with only two variables per constraint which is closely related to directed Diophantine approximation as well as the mixing set problem. We give a hardness result showing that the problem is NP-hard in general.

Motivated by the study of the mixing set problem and a recent result in the field of real-time systems we investigate the case of harmonic divisors, i.e. $a_{i+1}/a_i$ is an integer for all $i < n$. We present an algorithm to decide the feasibility of an instance in time $O(n^2)$ and we show that even the smallest feasible solution can be computed in strongly polynomial time $O(n^3)$.

1 Introduction

Integer programming is known as one of the most important fields in algorithm theory. This is due to the fact that a variety of problems actually can be modeled as an integer program. In the recent past there was a great interest in the so-called $n$-fold IPs [12] and 2-stage IPs [13]. The matrix $A$ of a 2-stage IP is constructed by blocks $A^{(1)}, \ldots, A^{(n)} \in \mathbb{Z}^{r \times k}$ and $B^{(1)}, \ldots, B^{(n)} \in \mathbb{Z}^{r \times t}$ as follows:

\[
A = \begin{pmatrix}
A^{(1)} & B^{(1)} & 0 & \cdots & 0 \\
A^{(2)} & 0 & B^{(2)} & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
A^{(n)} & 0 & \cdots & 0 & B^{(n)}
\end{pmatrix}
\]

For an objective vector $c \in \mathbb{Z}_{\geq 0}^{k+nt}$ and bounds $\ell, u \in \mathbb{Z}^{k+nt}$ the 2-stage IP is formulated as

\[
\max \{ c^T x | Ax = b, \ell \leq x \leq u, x \in \mathbb{Z}^{k+nt} \}.
\]

A special case of a 2-stage IP is given by the mixing set problem [2, 7, 14] (with only two variables in each constraint) where especially $r = k = t = 1$ and $A^{(1)} = \cdots = A^{(n)}$. Remark that the 2-variable integer programming problem was extensively studied by various authors, e.g. [2, 7, 14]. The mixing set problem plays an important role for example in integer programming approaches for production planning [17]. Given vectors $a, b \in \mathbb{Q}^n$ one aims to compute

\[
\min \{ f(s, x) | s + a_i x_i \geq b_i \forall i = 1, \ldots, n, (s, x) \in \mathbb{Z}_{\geq 0} \times \mathbb{Z}^n \}
\] (1)
for some objective function $f$. Conforti et al. \cite{5} pose the question whether the problem can be solved in polynomial time for linear functions $f$. Unless $P = NP$ this was ruled out by Eisenbrand and Rothvoß \cite{8} who proved that optimizing any linear function over a mixing set is NP-hard. However, the problem can be solved in polynomial time if $a_i = 1$ \cite{10, 15} or if the capacities $a_i$ fulfil a harmonic property \cite{5, 6, 18}, i.e. $a_{i+1}/a_i$ is integer for all $i < n$.

Now a recent manuscript in the field of real-time systems by Nguyen et al. \cite{16} gives rise to the study of a new problem. They present an exact algorithm for the worst-case response time analysis of harmonic tasks with constrained release jitter running in polynomial time. Their algorithm uses heuristic components to solve an integer program that can be stated as a bounded version of the mixing set problem with additional upper bounds $B_i$ as follows.

**Bounded Mixing Set (BMS)**

Given capacities $a_1, \ldots, a_n \in \mathbb{Z}$ and bounds $b, B \in \mathbb{Z}^n$ find $(s, x) \in \mathbb{Z}_{\geq 0} \times \mathbb{Z}^n$ such that

\[
b_i \leq s + a_i x_i \leq B_i \quad \forall i = 1, \ldots, n.
\]

In particular they depend on minimizing the value of $s$. This can be achieved in linear time in case of the original mixing set. See Lemma \cite{17} in the appendix for the short proof. While BMS may look artificial at first sight it is not; in fact, leading to a very natural generalization it can be restated in the well-known form of *simultaneous congruences*.

**Fuzzy Simultaneous Congruences (FSC)**

Given divisors $a_1, \ldots, a_n \in \mathbb{Z} \setminus \{0\}$ and remainder intervals $R_1, \ldots, R_n \subseteq \mathbb{Z}$ and an interval $S \subseteq \mathbb{Z}_{\geq 0}$ find a number $s \in S$ such that

\[
\exists r_i \in R_i : s \equiv r_i \pmod{a_i} \quad \forall i = 1, \ldots, n.
\]

Both problems BMS and FSC are interchangeable formulations of the same problem (see Section \ref{sec:contrib}). Therefore, we will use them as synonyms and we especially assume formally that $R_i = [b_i, B_i]$.

To the best of our knowledge, FSC/BMS was not considered before. However, the investigation of simultaneous congruences has always been of transdisciplinary interest connecting a variety of fields and applications, e.g. \cite{1, 9, 11}.

**Our Contribution.** We show that BMS is NP-hard for general capacities $a_i$. In the case of harmonic capacities, i.e. $a_{i+1}/a_i$ is an integer for $i < n$, we use a merge idea based on modular arithmetics for intervals to decide the feasibility problem of FSC in time $O(n^2)$. Furthermore, for a feasible instance of FSC we managed to compute the smallest feasible solution in strongly polynomial time, namely $O(\min\{n^2 \log(a_n), n^3\}) \leq O(n^3)$.

## 2 Hardness of BMS

We reduce from the problem of **Directed Diophantine Approximation** with *rounding down*. For any vector $v \in \mathbb{R}^n$ let $\lfloor v \rfloor$ denote the vector where each component is rounded down, i.e. $\lfloor v \rfloor_i = \lfloor v_i \rfloor$ for all $i \leq n$.

**Directed Diophantine Approximation with rounding down (DDA\textsuperscript{↓})**

Given: $a_1, \ldots, a_n \in \mathbb{Q}_+, N \in \mathbb{N}, \varepsilon \in \mathbb{Q}, 0 \leq \varepsilon < 1$ Decide whether there is a $Q \in \{1, \ldots, N\}$ such that $\|Qa - \lfloor Qa \rfloor\|_\infty \leq \varepsilon$.

Eisenbrand and Rothvoß proved that DDA\textsuperscript{↓} is NP-hard \cite{8} and actually, DDA\textsuperscript{↓} can be embedded
perfectly into a bounded mixing set, which yields the following theorem.

**Theorem 1.** BMS is NP-hard (even if $b_i = 0$ for all $i$ with $a_i \neq 0$).

**Proof.** Write $a_i = \beta_i/\gamma_i$ for integers $\beta_i \geq 0, \gamma_i \geq 1$ and set $\lambda = \prod_j \beta_j$. Then $\lambda/a_i = (\lambda/\beta_i)\gamma_i \geq 0$ is integer. Let $\mathcal{M}$ denote the following bounded mixing set:

\[
\begin{align*}
0 & \leq Q' - (\lambda/a_i) \cdot y_i & \leq \lfloor (\lambda/a_i) \cdot \varepsilon \rfloor & \forall i = 1, \ldots, n \quad (2) \\
\lambda & \leq Q' - 0 \cdot y_{n+1} & \leq \lambda \cdot N & \quad (3) \\
0 & \leq Q' - \lambda \cdot y_{n+2} & \leq 0 & \quad (4)
\end{align*}
\]

So let $Q \in \{1, \ldots, N\}$ with $\|Q\alpha - [Q\alpha]\|_\infty \leq \varepsilon$ be given. We obtain readily that $Q' = \lambda Q$ and $y = ([Q\alpha_1], \ldots, [Q\alpha_n], 0, Q)$ defines a solution of $\mathcal{M}$.

Vice-versa let $(Q', y)$ be a solution to $\mathcal{M}$. We see that (2) implies that $0 \leq Q' - (\lambda/a_i) \cdot y_i \leq \lfloor (\lambda/a_i) \cdot \varepsilon \rfloor \leq (\lambda/a_i) \cdot \varepsilon$ and by (3) we get that $Q' = \lambda \cdot y_{n+2}$ which then implies $0 \leq y_{n+2} \cdot a_i - y_i \leq \varepsilon < 1$ for all $i \leq n$.

Now, since $y_i$ is integer, there can be only one value for $y_i$, i.e. $y_i = [y_{n+2} a_i]$. By $Q' = \lambda \cdot y_{n+2}$ and (3) we get $y_{n+2} \in \{1, \ldots, N\}$ and setting $Q = y_{n+2}$ this yields $\|Q\alpha - [Q\alpha]\|_\infty \leq \varepsilon$ and that proves the claim.

\[\square\]

### 3 Notation and General Properties

For the sake of readability we write $X^{[\alpha]} = (X \mod \alpha)$ for numbers $X$ as well as $X^{[\alpha]} = \{z \mod \alpha \mid z \in X\}$ for sets $X$ (of numbers). Extending the usual notation we also write $X \equiv Y \pmod{\alpha}$ if $X^{[\alpha]} = Y^{[\alpha]}$ for sets $X, Y$. Remark that on the one hand $(X \cup Y)^{[\alpha]} = X^{[\alpha]} \cup Y^{[\alpha]}$ but on the other hand be aware that $(X \cap Y)^{[\alpha]} \neq X^{[\alpha]} \cap Y^{[\alpha]}$ in general (cf. Lemma 3).

Figure 1 depicts the structure of $v^{[\alpha]}$ if $v = [\ell_v, u_v]$ is an interval in $\mathbb{Z}$.

Also we use the well-tried notation $t + X = \{t + z \mid z \in X\}$ to express the translation of a set of numbers $X$ by some number $t$. For a set of sets $\mathcal{S}$ we write $\bigcup \mathcal{S}$ to denote the union $\bigcup_{S \in \mathcal{S}} S$. Furthermore, we identify constraints by their indices. So for $i \leq n$ we say that "$b_i \leq s + a_i x_i \leq b_i$" is constraint $i$.

![Diagram](image_url)

**Figure 1:** The two possibilities for the modular projection of an interval

**Identity of BMS and FSC.** In fact, BMS allows zero capacities while FSC cannot allow zero divisors since (mod 0) is undefined. However, suppose a constraint $i$ with $a_i \neq 0$. Let $b_i \leq s + a_i x_i \leq b_i$ and set $r_i = s + a_i x_i$. Then $r_i^{[\alpha]} = s^{[\alpha]}$ and $r_i \in [b_i, b_i] = R_i$. Vice-versa let $r_i \in R_i$ s.t. $r_i \equiv s \pmod{a_i}$. Then there is a $x_i \in \mathbb{Z}$ s.t. $s + a_i x_i = r_i$, $r_i \in [b_i, B_i]$.

A constraint $i$ that holds $a_i = 0$ simply requires that $s \in R_i$. Hence, if $a_i = a_j = 0$ for two constraints $i \neq j$ they can be replaced by one new constraint $k$ defined by $R_k = R_i \cap R_j$. Therefore,
one may assume that there is at most one constraint \( i \) with a zero capacity \( a_i \). As all our results can be lifted to the general case with low effort we will assume in terms of BMS that all capacities are non-zero and for FSC we take the equivalent assumption that \( S = Z_{>0} \).

With our notation we may easily express the feasibility of a value \( s \) for a single constraint \( i \) as follows.

**Observation 2.** A value \( s \) satisfies constraint \( i \) if and only if \( s^{[a_i]} \in R_i^{[a_i]} \).

**Proof.** \( \exists r_i \in R_i : r_i \equiv s \pmod{a_i} \iff \exists r_i \in R_i : r_i^{[a_i]} = s^{[a_i]} \iff s^{[a_i]} \in R_i^{[a_i]} \). \( \square \)

By simply swapping the signs of the \( x_i \) we may assume that \( a_i \geq 0 \) for all \( i \). We may also assume that the intervals are small in the sense that \( B_i - b_i + 1 < a_i \) holds for all \( i \). Assume that \( B_i - b_i + 1 \geq a_i \) for an \( i \) and let \( s \geq 0 \) be an arbitrary integer. Then \( b_i \leq B_i - a_i + 1 \) and constraint \( i \) may always be solved by setting \( x_i = \left\lfloor \frac{(b_i - s)}{a_i} \right\rfloor \) which yields

\[
\begin{align*}
b_i &\leq s + a_i \left\lfloor \frac{b_i - s}{a_i} \right\rfloor \leq s + a_i \left\lfloor \frac{B_i - a_i + 1 - s}{a_i} \right\rfloor = s + a_i \left\lfloor \frac{B_i - s}{a_i} \right\rfloor \leq B_i.
\end{align*}
\]

Hence, constraint \( i \) is redundant and may be omitted. As a direct consequence there can be at most one feasible value for each \( x_i \) for a given guess \( s \). In fact, we can decide the feasibility of a guess \( s \) in time \( O(n) \) as for all constraints \( i \) and values \( x_i \) it holds

\[
\begin{align*}
b_i &\leq s + a_i x_i \leq B_i \iff \left\lfloor \frac{(b_i - s)}{a_i} \right\rfloor = x_i = \left\lfloor \frac{(B_i - s)}{a_i} \right\rfloor.
\end{align*}
\]

So a guess \( s \) is feasible if and only if \( \left\lfloor \frac{(b_i - s)}{a_i} \right\rfloor = \left\lfloor \frac{(B_i - s)}{a_i} \right\rfloor \) holds for all constraints \( i \). By \( s_{\min} \) we denote the smallest feasible solution \( s \) that satisfies all constraints.

**Observation 3.** For feasible instances it holds that \( s_{\min} < \text{lcm}(a_1, \ldots, a_n) \).

**Proof.** Let \( \varphi = \text{lcm}(a_1, \ldots, a_n) \). Remark that \( \varphi/a_i \) is integral for all \( i \). Assume that \( (s, x) \) is a solution with \( s = s_{\min} \geq \varphi \). Let \( t = s - \varphi \) and \( y_i = x_i + \varphi/a_i \) f.a. \( i \). Then \( 0 \leq t < s_{\min} \) and \( t + a_i y_i = s + a_i x_i \) f.a. \( i \). So \( (t, y) \) is a solution that contradicts the optimality. \( \square \)

### 4 Harmonic Divisors

In the case of harmonic divisors it holds that \( a_{i+1}/a_i \) is an integer for all \( i < n \). Here we present an algorithm to decide the feasibility of an instance of FSC. Also we show how to compute the smallest feasible solution \( s_{\min} \) if it exists. For some intuition Figure 2 gives a perspective on \( s \) as an anchor for 1-dimensional lattices with basis \( a_i \) which have to hit the intervals \( R_i \). The idea for our first algorithm will be to decide the feasibility problem by iteratively computing modular projections from constraint \( i = n \) down to \( i = 1 \). Some more notation will be helpful.

In the following we will say that an interval \( w \subseteq Z \) represents a set \( M \subseteq Z \) (modulo \( \alpha \)) if \( w^\alpha = M^\alpha \). Also a set of intervals \( \mathcal{R} \) represents a set \( M \subseteq Z \) (modulo \( \alpha \)) if \( M^\alpha = \bigcup_{w \in \mathcal{R}} w^\alpha \).
Given an integer $\alpha \geq 1$ and two intervals $v$, $w$ we depend on the structure of the intersection $v[\alpha] \cap w[\alpha] \subseteq [0, \alpha)$. To express it let $v = [\ell_v, u_v]$, $w = [\ell_w, u_w]$ and we define the basic intervals

$$\varphi_\alpha(v, w) = [\ell_v[\alpha], u_w[\alpha]] \quad \text{and} \quad \psi_\alpha(v, w) = \max\{\ell_v[\alpha], \ell_w[\alpha]\}, \alpha + \min\{u_v[\alpha], u_w[\alpha]\}$$

for all intervals $v$, $w$. Remark that $\psi_\alpha(v, w) = \psi_\alpha(v, w)$ is always true.

**Lemma 4.** Given an integer $\alpha \geq 1$ and two intervals $v, w \subseteq \mathbb{Z}$ it holds that

$$v[\alpha] \cap w[\alpha] \in \{\emptyset, v[\alpha], w[\alpha], \psi_\alpha(v, w)[\alpha], \varphi_\alpha(v, w), \varphi_\alpha(v, w) \cup \psi_\alpha(v, w)[\alpha] \}$$

The important intuition is that such a “modulo $\alpha$ intersection” can always be represented by at most two intervals. Remark that the sets in the second row are the only ones which are represented by $2 > 1$ intervals.

**Proof.** We do a case distinction (see Figure 3) as follows. We only look at the non-trivial case, i.e. $v[\alpha] \cap w[\alpha] \notin \{\emptyset, v[\alpha], w[\alpha]\}$, which especially implies $|v| < \alpha$ and $|w| < \alpha$.

We start with the case that neither $v[\alpha]$ nor $w[\alpha]$ is an interval, i.e. $u_v[\alpha] < \ell_v[\alpha]$ and $u_w[\alpha] < \ell_w[\alpha]$. Then it cannot be that $u_w[\alpha] \geq \ell_v[\alpha]$ and $u_v[\alpha] \geq \ell_w[\alpha]$ since that implies $\ell_v[\alpha] \leq u_w[\alpha] < \ell_w[\alpha] \leq u_v[\alpha]$. Hence, there are three cases as follows.

**Case 1.** $u_w[\alpha] < \ell_v[\alpha]$ and $u_v[\alpha] < \ell_w[\alpha]$. Then the intersection equals

$$[0, \min\{u_v[\alpha], u_w[\alpha]\}] \cup [\max\{\ell_v[\alpha], \ell_w[\alpha]\}, \alpha] = [\max\{\ell_v[\alpha], \ell_w[\alpha]\}, \alpha + \min\{u_v[\alpha], u_w[\alpha]\}] = \psi_\alpha(v, w)[\alpha].$$

**Case 2.** $u_v[\alpha] \geq \ell_v[\alpha]$ and $u_v[\alpha] < \ell_w[\alpha]$. Then the intersection equals

$$[0, \min\{u_v[\alpha], u_w[\alpha]\}] \cup [\ell_v[\alpha], u_w[\alpha]] \cup [\ell_w[\alpha], \alpha] = [\ell_v[\alpha], u_v[\alpha]] \cup [\ell_w[\alpha], \alpha + u_v[\alpha]] = \varphi_\alpha(v, w) \cup \psi_\alpha(v, w)[\alpha].$$

**Case 3.** $u_v[\alpha] < \ell_v[\alpha]$ and $u_v[\alpha] \geq \ell_w[\alpha]$. By symmetry we get $v[\alpha] \cap w[\alpha] = \varphi_\alpha(v, w) \cup \psi_\alpha(v, w)[\alpha]$. Now, w.l.o.g. assume that $v[\alpha]$ is an interval, i.e. $\ell_v[\alpha] \leq u_v[\alpha]$, while $w[\alpha]$ consists of two intervals, i.e. $u_w[\alpha] < \ell_w[\alpha]$. Then there are three cases as follows.

**Case 2.1.** $\ell_v[\alpha] \leq u_w[\alpha] < u_v[\alpha] < \ell_w[\alpha]$. Then the intersection equals $[\ell_v[\alpha], u_w[\alpha]] = \varphi_\alpha(v, w)$.

**Case 2.2.** $u_v[\alpha] < \ell_v[\alpha] < \ell_w[\alpha] \leq u_w[\alpha]$. Then the intersection equals $[\ell_v[\alpha], \ell_w[\alpha]] = \varphi_\alpha(v, w)$.

**Case 2.3.** $\ell_v[\alpha] \leq u_w[\alpha] < \ell_w[\alpha] \leq u_v[\alpha]$. Then the intersection is

$$[\ell_v[\alpha], u_w[\alpha]] \cup [\ell_v[\alpha], u_v[\alpha]] = \varphi_\alpha(v, w) \cup \varphi_\alpha(v, w).$$

Clearly, if both $v[\alpha]$ and $w[\alpha]$ are intervals (Case 3) (which are not disjoint) then their intersection is either $\varphi_\alpha(v, w)$ or $\varphi_\alpha(v, w)$.

\[\square\]
Lemma 5. Let $\alpha \geq 1$, let $v$ be an interval and let $Q$ be a set of $k \geq 1$ intervals. Then there is a set $R$ of at most $k + 1$ intervals such that

$$v^{[\alpha]} \cap \left( \bigcup Q \right)^{[\alpha]} = \left( \bigcup R \right)^{[\alpha]}.$$ 

Carefully remark that this is not the same as “$v \cap \bigcup Q \equiv \bigcup R$ (mod $\alpha$)” because of the intersection. We will now give the proof.

Proof. We simply obtain that

$$v^{[\alpha]} \cap \left( \bigcup Q \right)^{[\alpha]} = \bigcup_{w \in Q} (v^{[\alpha]} \cap w^{[\alpha]}) = \bigcup_{w \in D} (v^{[\alpha]} \cap w^{[\alpha]})$$

where $D = \{ w \in Q \mid w^{[\alpha]} \subseteq v^{[\alpha]} \}$. We set $\alpha = \min \{ v \cap w \mid w \in D \}$.

In order to get an upper bound we assume that these two types of intersections do not come together. In more detail, we may assume by symmetry that $D = D_1 \cup D_2$ where

$$D_1 = \{ w \in D \mid v^{[\alpha]} \cap w^{[\alpha]} = \varphi_\alpha(v, w) \} \quad \text{and} \quad D_2 = \{ w \in D \mid v^{[\alpha]} \cap w^{[\alpha]} = \varphi_\alpha(v, w) \}.$$ 

It turns out that

$$\bigcup_{w \in D_1} (v^{[\alpha]} \cap w^{[\alpha]}) = \bigcup_{w \in D_1} ([[\ell_v^{[\alpha]} - v^{[\alpha]}], [\ell_v^{[\alpha]} + v^{[\alpha]}]]) = [\ell_v^{[\alpha]}, \max_{w \in D_1} \ell_v^{[\alpha]}]$$

and

$$\bigcup_{w \in D_2} (v^{[\alpha]} \cap w^{[\alpha]}) = \bigcup_{w \in D_2} ([[\ell_v^{[\alpha]} - v^{[\alpha]}], [\ell_v^{[\alpha]} + v^{[\alpha]}]] \cap [\ell_v^{[\alpha]} + v^{[\alpha]}])$$

which finally joins up to

$$\bigcup_{w \in D} (v^{[\alpha]} \cap w^{[\alpha]}) = [\ell_v^{[\alpha]}, \max_{w \in D} \ell_v^{[\alpha]} \cap \min_{w \in D} \ell_v^{[\alpha]}].$$

Hence, all intersections with intervals in $D$ may be represented by at most two intervals in total while each other intersection can be represented by at most one interval. Thus, if $|D| = 0$ then the whole intersection can be represented by at most $k$ intervals. If $|D| \geq 1$ then there are at most $2 + |Q| - |D| \leq 2 + k - 1 = k + 1$ intervals required.

Let $S_i$ denote the set of all solutions $s \in \mathbb{Z}_{\geq 0}$ that are feasible for each of the constraints $i, i + 1, \ldots, n$. We set $S_{n+1} = \mathbb{Z}_{\geq 0}$ to denote the feasible solutions to an empty set of constraints. The correctness of Algorithm 3 is implied by the following fundamental lemma. See Figure 4 for an example of a step inside the algorithm.
Obviously it holds that \( (\bigcup Q_i)^{[a_i]} = R_i^{[a_i]} \cap (\bigcup Q_{i+1})^{[a_i]} \) and \( Q_i \leq O(n-i) \).

**Lemma 6.** It holds true that \( S_i^{[a_i]} = R_i^{[a_i]} \cap S_{i+1}^{[a_i]} \) for all \( i = 1, \ldots, n \).

**Proof.** Let \( r \in S_i^{[a_i]} \). So there is a solution \( s \in S_i \) such that \( r = s^{[a_i]} \in R_i^{[a_i]} \). It holds that \( S_i \subseteq S_{i+1} \) which implies \( s \in S_{i+1} \) and thus \( r = s^{[a_i]} \in S_{i+1}^{[a_i]} \).

Vice-versa let \( r = R_i^{[a_i]} \cap S_{i+1}^{[a_i]} \). So there is a solution \( s \in S_{i+1} \) with \( s^{[a_i]} = r \). From \( r \in R_i^{[a_i]} \) we get \( s^{[a_i]} \in R_{i+1}^{[a_i]} \). Hence, \( s \in S_i \) and \( r = s^{[a_i]} \in S_i^{[a_i]} \).

**Theorem 7.** Algorithm 1 decides the feasibility of an instance in time \( O(n^2) \).

**Proof.** We show that \( \bigcup Q_1 = S_i \) (mod \( a_i \)) for all \( i = n, \ldots, 1 \). This will prove the algorithm correct since then \( \bigcup Q_1 \equiv S_1 \) (mod \( a_1 \)) and that means \( \bigcup Q_1 \) is empty if and only if \( S_1 \) is empty. Obviously it holds that \( \bigcup Q_n = S_n \) (mod \( a_n \)) since \( \bigcup Q_n = R_n \). Now suppose that \( \bigcup Q_{i+1} \equiv S_{i+1} \) (mod \( a_{i+1} \)) for some \( i \geq 1 \). We have that

\[
(\bigcup Q_i)^{[a_i]} = R_i^{[a_i]} \cap (\bigcup Q_{i+1})^{[a_i]}
\]

and \( a_i | a_{i+1} \) implies \( (\bigcup Q_{i+1})^{[a_i]} = ((\bigcup Q_{i+1})^{[a_{i+1}]})^{[a_i]} = (S_{i+1}^{[a_{i+1}]})^{[a_i]} = S_{i+1}^{[a_i]} \). With Lemma 6 this yields

\[
(\bigcup Q_i)^{[a_i]} = R_i^{[a_i]} \cap S_{i+1}^{[a_i]} = S_i^{[a_i]}
\]

and that proves the algorithm correct. Using Lemmas 4 to 6 each set \( Q_i \) can be computed in time \( O(n) \) and this yields a total running time of \( O(n^2) \).

**Corollary 8.** For feasible instances \( s_{\min} \) can be computed in time \( O(n^2 \log(a_n)) \).

This can be achieved by introducing an additional constraint measuring the value of \( s \) as follows. Let \( \beta \) be a positive integer. We extend the problem instance by a new constraint with number \( n+1 \) defined by

\[
a_{n+1} = 2 \cdot a_n, \quad b_{n+1} = 0, \quad B_{n+1} = \beta.
\]
Remark that this \( \beta \)-instance admits the same set of solutions as the original instance as long as \( \beta \) is large enough, e.g. \( \beta = a_n \) (cf. Observation 3). Consider a feasible solution to the \( \beta \)-instance where \( \beta \leq a_n \). It holds that

\[
2a_n x_{n+1} = a_{n+1} x_{n+1} + s + a_{n+1} x_{n+1} \leq B_{n+1} = \beta \leq a_n
\]

which implies \( x_{n+1} \leq \lfloor \frac{B_{n+1}}{2a_n} \rfloor = 0 \). However, if \( x_{n+1} < 0 \) then \( s \geq a_{n+1} \cdot |x_{n+1}| \) and therefore the solution \( s' = s + a_{n+1} x_{n+1} \) with \( x_{n+1} = 0 \) and \( x_i' = x_i - (a_{n+1}/a_i)x_{n+1} \) for all \( i = 1, \ldots, n \) is better than \( s \) and \( x_{n+1}' = 0 \).

Thus we may assume generally that \( x_{n+1} = 0 \) which allows us to measure the value of \( s \) using the upper bound \( \beta \). We use \( \beta \) to do a binary search in the interval \([0, a_n]\) using Algorithm 4 to check the \( \beta \)-instance for feasibility. The smallest possible value for \( \beta \) then states the optimum value and that proves Corollary 8.

However, with additional ideas we are able to achieve strongly polynomial time. The next lemma seems to be a key property of modular arithmetics on sets.

**Lemma 9.** For all numbers \( a, b \in \mathbb{Z}_{\geq 1} \) and sets \( A, B \subseteq \mathbb{Z} \) it holds

\[
A^{[a]} \cap B^{[a]} = \left( A^{[ab]} \cap \bigcup_{i=0}^{b-1} (ia + B^{[a]}) \right)^{[a]}
\]

**Proof.** Let \( x \) be a number. Then it holds

\[
x \in \left( A^{[ab]} \cap \bigcup_{i=0}^{b-1} (ia + B^{[a]}) \right)^{[a]} \iff \exists y \in A^{[ab]} : y \in \bigcup_{i=0}^{b-1} (ia + B^{[a]}) \wedge x = y^{[a]}
\]

\[
\iff \exists y \in A^{[ab]} : y^{[a]} \in B^{[a]} \wedge x = y^{[a]}
\]

\[
\iff x \in A^{[a]} \cap B^{[a]}
\]

where the last equivalence follows from \( (A^{[ab]})^{[a]} = A^{[a]} \). \( \Box \)

Since the right side can be written as the modular projection of a union of intersections modulo \( a \) we can find a fine-grained strengthening; in fact, for arbitrary sets \( X, M_0, \ldots, M_{m-1} \) it holds that

\[
\bigcup_{i=0}^{m-1} (X \cap M_i) = \bigcup_{i=0}^{m-1} (X \cap (M_i \setminus \bigcup_{j=0}^{i-1}(X \cap M_j))).
\]

While the left-hand side may not, the right-hand side is always a disjoint union. Taking into account the modular projections this leads to the following corollary.

**Corollary 10.** For all numbers \( a, b \in \mathbb{Z}_{\geq 1} \) and sets \( A, B \subseteq \mathbb{Z} \) it holds

\[
A^{[a]} \cap B^{[a]} = \left( \bigcup_{i=0}^{b-1} D_i \right)^{[a]}
\]

where \( D_i = A^{[ab]} \cap Y_i \) and \( Y_i = ia + (B^{[a]} \setminus \bigcup_{j=0}^{i-1} D_j^{[a]}) \) for all \( i = 0, \ldots, b-1 \).

**Observation 11.** For feasible instances it holds that \( s_{\text{min}} \in R_n^{[a_n]} \).

This is true since in the harmonic case \( s_{\text{min}} < \text{lcm}(a_1, \ldots, a_n) = a_n \) due to Observation 3 which then implies that \( s_{\text{min}} = s_{\text{min}}^{[a_n]} \in R_n^{[a_n]} \) using Observation 2.

The idea is to search for \( s_{\text{min}} \) in the modular projection \( R_n^{[a_n]} \) by aggregating the penultimate constraint \( n-1 \) into the last constraint \( n \). Fortunately, the number of intervals needed to represent both constraints can be bounded by a constant. A fine-grained construction then enforces the algorithm to efficiently iterate the feasibility test on aggregated instances to find the optimum value.
Theorem 12. For feasible instances $s_{\text{min}}$ can be computed in time $O(n^3)$.

Remark that the set of feasible solutions for the last two constraints is $S_{n-1} = R_{n-1}^{[a_{n-1}]} \cap (R_n^{[a_n]})_{a_{n-1}} = R_{n-1}^{[a_{n-1}]} \cap R_n^{[a_{n-1}]}$. Therefore, the next lemma states the crucial argument of the algorithm.

Lemma 13. The intersection $R_{n-1}^{[a_{n-1}]} \cap R_n^{[a_{n-1}]}$ can always be represented by the disjoint union $U \subseteq R_n^{[a_n]}$ of only constant many intervals in $R_n^{[a_n]}$ such that

(a) $U_{[a_{n-1}]} = R_{n-1}^{[a_{n-1}]} \cap R_n^{[a_{n-1}]}$ and \hspace{1cm} (Representation)

(b) $u \equiv r \pmod{a_{n-1}}$ implies $u \leq r$ for all $u, r \in R_n^{[a_n]}$. \hspace{1cm} (Minimality)

Here the former property states that indeed the intervals in $U$ are a proper representation for the last two constraints. The important property is the latter; in fact, it ensures that $U$ is the best possible representation in the sense that $U$ consists of the smallest intervals possible (see Figure 5).

Proof of Lemma 13 (a). By defining $D_i = Y_i \cap R_n^{[a_n]}$ and

$$Y_i = ia_{n-1} + \left( R_{n-1}^{[a_{n-1}]} \setminus \bigcup_{j=0}^{i-1} D_j^{[a_{n-1}]} \right)$$

for all $i \in \{0, \ldots, a_n/a_{n-1} - 1\}$ Corollary 14 proves the claim (cf. Figure 5). (b) follows by construction.

It remains to show that $\bigcup D_i$ is the union of only constant many disjoint intervals. Apparently, the intervals are disjoint by construction.

We claim that there are at most three non-empty sets $D_i$. Assume there are at least four non-empty translates $D_i$, namely $D_1, D_2, D_3, D_4$. Then, since $R_n$ is an interval it holds for at least two $p, q \in \{i, j, k, \ell\}$ that the full interval translates $F_p = [pa_{n-1}, (p+1)a_{n-1}]$ and $F_q = [qa_{n-1}, (q+1)a_{n-1}]$ are subsets of $R_n^{[a_n]}$. For $p$ (and also for $q$) we get

$$D_p^{[a_{n-1}]} = (\bigcup_{Y \subseteq F_p} Y \cap R_n^{[a_n]})_{[a_{n-1}]} = Y_p^{[a_{n-1}]} = R_{n-1}^{[a_{n-1}]} \setminus \bigcup_{j=0}^{p-1} D_j^{[a_{n-1}]}$$

which implies with $\bigcup_{j=0}^{p-1} D_j^{[a_{n-1}]} \subseteq R_{n-1}^{[a_{n-1}]}$ that

$$\bigcup_{j=0}^{p} D_j^{[a_{n-1}]} = D_p^{[a_{n-1}]} \cup \bigcup_{j=0}^{p-1} D_j^{[a_{n-1}]} = R_{n-1}^{[a_{n-1}]}$$

Figure 5: An example for four required intervals to represent $R_{n-1}^{[a_{n-1}]} \cap R_n^{[a_{n-1}]}$ in Lemma 13.
Then it follows \( \bigcup_{j=q}^{p} D_j^{[a_{n-1}]} = R_{[a_{n-1}]}^{[a_{n-1}]} = \bigcup_{j=q}^{p} D_j^{[a_{n-1}]} \). W.l.o.g. let \( p < q \). Then \( D_q = Y_q \cap R_{[a_{n-1}]}^{[a_{n-1}]} \) is empty since

\[
Y_q = qa_{n-1} + \left( R_{[a_{n-1}]}^{[a_{n-1}]} \setminus \bigcup_{j=0}^{q-1} D_j^{[a_{n-1}]} \right) \subseteq qa_{n-1} + \left( R_{[a_{n-1}]}^{[a_{n-1}]} \setminus R_{[a_{n-1}]}^{[a_{n-1}]} \right)
\]
is empty and we have a contradiction.

Using the same case distinctions as in the proof of Lemma 4 one can show that each set \( D_i \) consist of at most two intervals. Therefore, all the non-empty sets \( D_i \) consist of at most \( 3 \cdot 2 = 6 \) intervals in total. In fact, one can improve this bound to a total number of at most 4 intervals (see Figure 5) by a more sophisticated case distinction.

This admits an algorithm using an aggregation argument as follows. For constraints \( n \) and \( n-1 \) we use Lemma 13 to compute disjoint intervals \( E_1, \ldots, E_k \subseteq R_{[a_{n-1}]}^{[a_{n-1}]} \) (representing the constraints \( n \) and \( n-1 \)) where \( k \leq C \) for a small constant \( C \). If \( k \geq 1 \) then use Algorithm 1 to check the feasibility of the instances \( I_1, \ldots, I_k \) defined by

\[
(I_j) \quad \min s \\
\quad s^{[a_i]} \in R_{[a_i]}^{[a_i]} \quad \forall i = 1, \ldots, n-2 \\
\quad s^{[a_n]} \in E_j \\
\quad s \in \mathbb{Z}_{\geq 0}.
\]

If none of the instances \( I_1, \ldots, I_k \) admits a solution then the original instance can not be feasible. Assume that there is at least one feasible instance. Now, since \( E_1, \ldots, E_k \) are disjoint exactly one of them contains the optimum value for \( s \). W.l.o.g. assume that \( E_1 < \cdots < E_k \). Then there is a smallest index \( j \) such that \( I_j \) is feasible and we solve \( I_j \) recursively to find the optimum value. Together this yields an algorithm running in time \( n \cdot C \cdot O(n^2) = O(n^3) \).

References

[1] Manindra Agrawal and Somenath Biswas. Primality and identity testing via chinese remaindering. In Proc. FOCS 1999, pages 202–209, 1999.

[2] Reuven Bar-Yehuda and Dror Rawitz. Efficient algorithms for integer programs with two variables per constraint. Algorithmica, 29(4):595–609, 2001.

[3] Michele Conforti, Gerard Cornuéjols, and Giacomo Zambelli. Integer Programming. Springer Publishing Company, Incorporated, 2014.

[4] Michele Conforti, Marco Di Summa, and Laurence A. Wolsey. The mixing set with flows. SIAM J. Discrete Math., 21(2):396–407, 2007.

[5] Michele Conforti, Marco Di Summa, and Laurence A. Wolsey. The mixing set with divisible capacities. In Proc. IPCO 2008, pages 435–449, 2008.

[6] Michele Conforti and Giacomo Zambelli. The mixing set with divisible capacities: A simple approach. Oper. Res. Lett., 37(6):379–383, 2009.

[7] Friedrich Eisenbrand and Günter Rote. Fast 2-variable integer programming. In Proc. IPCO 2001, pages 78–89, 2001.

[8] Friedrich Eisenbrand and Thomas Rothvoß. New hardness results for diophantine approximation. In Proc. APPROX 2009, pages 98–110, 2009.
A Smallest Feasible $s$ for Mixing Set

Lemma 14. For $f(s, x) = s$ the mixing set (1) can be solved in linear time.

Proof. We show that $s_{\text{min}} = s^* := \max(\{ 0 \} \cup \{ b_i \mid a_i = 0 \})$ where $s_{\text{min}}$ denotes the optimal solution to (1) for $f(s, x) = s$. Let $i \leq n$.

Case $s^* \geq b_i$. Set $x_i^* = 0$. Then we have $s^* + a_i x_i^* = s^* \geq b_i$.

Case $s^* < b_i$. Then $a_i \neq 0$ and $b_i - s^* > 0$. We set $x_i^* = \left\lfloor \frac{1}{a_i} (b_i - s^*) \right\rfloor$ if $a_i > 0$ and $x_i^* = \left\lceil \frac{1}{a_i} (b_i - s^*) \right\rceil$ if $a_i < 0$. Again we get that $s^* + a_i x_i^* \geq b_i$.

Hence, $s^*$ is a solution. Apparently $s^*$ is optimal if $s^* = 0$. If $s^* > 0$ then there is a constraint $j$ with $a_j = 0$ such that $s^* = b_j \leq s_{\text{min}} + a_j x_j = s_{\text{min}}$ for any $x_j$. \qed