On the String Consensus Problem and the Manhattan Sequence Consensus Problem

Tomasz Kociumaka[1,⋆], Jakub W. Pachocki[2], Jakub Radoszewski[1,⋆⋆], Wojciech Rytter[1,3], and Tomasz Waleń[1]

[1] Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland
{kociumaka,jrad,rytter,walen}@mimuw.edu.pl
[2] Carnegie Mellon University
pachocki@cs.cmu.edu
[3] Faculty of Mathematics and Computer Science, Copernicus University, Toruń, Poland

Abstract. In the Manhattan Sequence Consensus problem (MSC problem) we are given k integer sequences, each of length ℓ, and we are to find an integer sequence $\mathbf{x}$ of length ℓ (called a consensus sequence), such that the maximum Manhattan distance of $\mathbf{x}$ from each of the input sequences is minimized. For binary sequences Manhattan distance coincides with Hamming distance, hence in this case the string consensus problem (also called string center problem or closest string problem) is a special case of MSC. Our main result is a practically efficient $\mathcal{O}(\ell)$-time algorithm solving MSC for $k \leq 5$ sequences. Practicality of our algorithms has been verified experimentally. It improves upon the quadratic algorithm by Amir et al. (SPIRE 2012) for string consensus problem for $k = 5$ binary strings. Similarly as in Amir's algorithm we use a column-based framework. We replace the implied general integer linear programming by its easy special cases, due to combinatorial properties of the MSC for $k \leq 5$. We also show that for a general parameter k any instance can be reduced in linear time to a kernel of size $k!$, so the problem is fixed-parameter tractable. Nevertheless, for $k \geq 4$ this is still too much for any naive solution to be feasible in practice.

1 Introduction

In the sequence consensus problems, given a set of sequences of length ℓ we are searching for a new sequence of length ℓ which minimizes the maximum distance to all the given sequences in some particular metric. Finding the consensus sequence is a tool for many clustering algorithms and as such has applications in unsupervised learning, classification, databases, spatial range searching, data mining etc [4]. It is also one of popular methods for detecting data commonalities of many strings (see [1]) and has a considerable number of applications

⋆ Supported by Polish budget funds for science in 2013-2017 as a research project under the 'Diamond Grant' program.
⋆⋆ The author receives financial support of Foundation for Polish Science.

E. Moura and M. Crochemore (Eds.): SPIRE 2014, LNCS 8799, pp. 244–255, 2014.

in coding theory [6,8], data compression [11] and bioinformatics [12,16]. The consensus problem has previously been studied mainly in $\mathbb{R}^\ell$ space with the Euclidean distance and in Σ^ℓ (that is, the space of sequences over a finite alphabet Σ) with the Hamming distance. Other metrics were considered in [2]. We study the sequence consensus problem for Manhattan metric (ℓ_1 norm) in correlation with the Hamming-metric variant of the problem.

The Euclidean variant of the sequence consensus problem is also known as the bounding sphere, enclosing sphere or enclosing ball problem. It was initially introduced in 2 dimensions (i.e., the smallest circle problem) by Sylvester in 1857 [22]. For an arbitrary number of dimensions, several approximation algorithms [4,15,21] and practical exact algorithms [7,10] have been proposed.

The Hamming-distance variant of the sequence consensus problem is known under the names of string consensus, center string or closest string problem. The problem is known to be NP-complete even for binary alphabet [8]. The algorithmic study of Hamming string consensus (HSC) problem started in 1999 with the first approximation algorithms [16]. Afterwards polynomial-time approximation schemes (PTAS) with different running times were presented [3,19,20]. A number of exact algorithms have also been proposed. Many of these consider a decision version of the problem, in which we are to check if there is a solution to HSC problem with distance at most d to the input sequences. Thus FPT algorithms with time complexities $\mathcal{O}(k\ell + kd^{d+1})$ and $\mathcal{O}(k\ell + kd(16|\Sigma|)^d)$ were presented in [12] and [19], respectively.

An FPT algorithm parameterized only by k was given in [12]. It uses Lenstra's algorithm [17] for a solution of an integer linear program of size exponential in k (which requires $\mathcal{O}(k!^{4.5k!}\ell)$ operations on integers of magnitude $\mathcal{O}(k!^{2k!}\ell)$, see [1]) and due to extremely large constants is not feasible for $k \geq 4$. This opened a line of research with efficient algorithms for small constant k. A linear-time algorithm for $k = 3$ was presented in [12], a linear-time algorithm for $k = 4$ and binary alphabet was given in [5], and recently an $\mathcal{O}(\ell^2)$-time algorithm for $k = 5$ and also binary alphabet was developed in [1].

For two sequences $\mathbf{x} = (x_1, \ldots, x_\ell)$ and $\mathbf{y} = (y_1, \ldots, y_\ell)$ the Manhattan distance (also known as rectilinear or taxicab distance) between $\mathbf{x}$ and $\mathbf{y}$ is defined as follows:

$$dist(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{\ell} |x_j - y_j|.$$

The Manhattan version of the consensus problem is formally defined as follows:

Manhattan Sequence Consensus **problem**

Input: A collection $\mathcal{A}$ of k integer sequences $\mathbf{a}_i$, each of length ℓ;

Output: $\mathrm{OPT}(\mathcal{A}) = \min_{\mathbf{x}} \max \{dist(\mathbf{x}, \mathbf{a}_i) : 1 \leq i \leq k\}$, and the corresponding integer consensus sequence $\mathbf{x}$.

We assume that integers $a_{i,j}$ satisfy $|a_{i,j}| \leq M$, and all ℓ, k, M fit in a machine word, so that arithmetics on integers of magnitude $\mathcal{O}(\ell M)$ take constant time.

For simplicity in this version of the paper we concentrate on computing $\mathrm{OPT}(\mathcal{A})$ and omit the details of recovering the corresponding consensus sequence $\mathbf{x}$. Nevertheless, this step is included in the implementation provided.

Example 1. Let $\mathcal{A} = ((120, 0, 80),\ (20, 40, 130),\ (0, 100, 0))$. Then $\mathrm{OPT}(\mathcal{A}) = 150$ and a consensus sequence is $\mathbf{x} = (30, 40, 60)$, see also Fig. 1.

Our results are the following:

- We show that Manhattan Sequence Consensus problem has a kernel with $\ell \leq k!$ and give an algorithm which works in linear time for any fixed k.
- We present a practical linear-time algorithm for the Manhattan Sequence Consensus problem for $k = 5$ (which obviously can be used for any $k \leq 5$).

Note that binary HSC problem is a special case of MSC problem. Hence, the latter problem is NP-complete. Moreover, the efficient linear-time algorithm presented here for MSC problem for $k = 5$ yields an equally efficient linear-time algorithm for the binary HSC problem and thus improves the result of [1].

Organization of the Paper. Our approach is based on a reduction of the MSC problem to instances of integer linear programming (ILP). For general constant k we obtain a constant, though a very large, number of instances with a constant number of variables that we solve using Lenstra's algorithm [17] which works in constant time (the constant coefficient of this algorithm is also very large). This idea is similar to the one used in the FPT algorithm for HSC problem [12], however for MSC it requires an additional combinatorial observation. For $k \leq 5$ we obtain a more efficient reduction of MSC to at most 20 instances of very special ILP which we solve efficiently without applying a general ILP solver.

In Section 2 we show the first steps of the reduction of MSC to ILP. In Section 3 we show a kernel for the problem of $\mathcal{O}(k!)$ size. In Section 4 we perform a combinatorial analysis of the case $k = 5$ which leaves 20 simple types of the sequence $\mathbf{x}$ to be considered. This analysis is used in Section 5 to obtain 20 special ILP instances with only 4 variables. They could be solved using Lenstra's ILP solver. However, there exists an efficient algorithm tailored for this type of special instances. Due to space constraints, it is omitted in this version; it can be found in [14]. Finally we analyze the performance of a C++ implementation of our algorithm in the Conclusions (Section 6).

2 From MSC Problem to ILP

Let us fix a collection $\mathcal{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_k)$ of the input sequences. The elements of $\mathbf{a}_i$ are denoted by $a_{i,j}$ (for $1 \leq j \leq \ell$). We also denote $dist(\mathbf{x}, \mathcal{A}) = \max\{dist(\mathbf{x}, \mathbf{a}_i) : 1 \leq i \leq k\}$.

For $j \in \{1, \ldots, n\}$ let π_j be a permutation of $\{1, \ldots, k\}$ such that $a_{\pi_j(1),j} \leq \ldots \leq a_{\pi_j(k),j}$, i.e. π_j is the ordering permutation of elements $a_{1,j}, \ldots, a_{k,j}$. We also set $s_{i,j} = a_{\pi_j(i),j}$, see Example 2. For some j there might be several possibilities for π_j (if $a_{i,j} = a_{i',j}$ for some $i \neq i'$), we fix a single choice for each j.

Example 2. Consider the following three sequences $\mathbf{a}_i$ and sequences $\mathbf{s}_i$ obtained by sorting columns:

$$[a_{i,j}] = \begin{bmatrix} 120 & 0 & 80 \\ 20 & 40 & 130 \\ 0 & 100 & 0 \end{bmatrix}, \quad [s_{i,j}] = \begin{bmatrix} 0 & 0 & 0 \\ 20 & 40 & 80 \\ 120 & 100 & 130 \end{bmatrix}.$$

The Manhattan consensus sequence is $\mathbf{x} = (30, 40, 60)$, see Fig. 1. In the figure, the circled numbers in j-th column are $\pi_j(1), \pi_j(2), \ldots, \pi_j(k)$ (top-down).

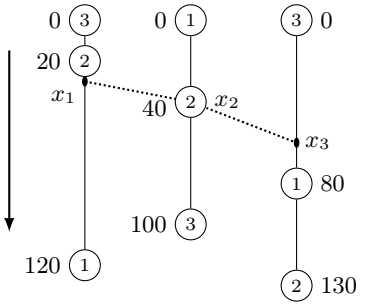

Fig. 1. Illustration of Example 2; $\pi_1 = (3, 2, 1)$, $\pi_2 = (1, 2, 3)$, $\pi_3 = (3, 1, 2)$

Definition 3. *A* basic interval *is an interval of the form* $[i, i+1]$ *(for* $i = 1, \ldots, k-1$*) or* $[i, i]$ *(for* $i = 1, \ldots, k$*). The former is called* proper*, and the latter* degenerate*. An* interval system *is a sequence* $\mathcal{I} = (I_1, \ldots, I_\ell)$ *of basic intervals* I_j*.*

For a basic interval I_j we say that a value x_j is consistent with I_j if $x_j \in \{s_{i,j}, \ldots, s_{i+1,j}\}$ when $I_j = [i, i+1]$ is proper, and if $x_j = s_{i,j}$ when $I_j = [i, i]$ is degenerate. A sequence $\mathbf{x}$ is called consistent with an interval system $\mathcal{I} = (I_j)_{j=1}^{\ell}$ if for each j the value x_j is consistent with I_j.

For an interval system $\mathcal{I}$ we define $\mathrm{OPT}(\mathcal{A}, \mathcal{I})$ as the minimum $dist(\mathbf{x}, \mathcal{A})$ among all integer sequences $\mathbf{x}$ consistent with $\mathcal{I}$. Due to the following trivial observation, for every $\mathcal{A}$ there exists an interval system $\mathcal{I}$ such that $\mathrm{OPT}(\mathcal{A}) = \mathrm{OPT}(\mathcal{A}, \mathcal{I})$.

Observation 4. *If* $\mathbf{x}$ *is a Manhattan consensus sequence then for each* j*,* $s_{1,j} \leq x_j \leq s_{k,j}$*.*

Transformation of the Input to an ILP. Note that for all sequences $\mathbf{x}$ consistent with a fixed $\mathcal{I}$, the Manhattan distances $dist(\mathbf{x}, \mathbf{a}_i)$ can be expressed as $d_i + \sum_{j=1}^{\ell} e_{i,j} x_j$ with $e_{i,j} = \pm 1$. Thus, the problem of finding $\mathrm{OPT}(\mathcal{A}, \mathcal{I})$ can be formulated as an ILP, which we denote $\mathrm{ILP}(\mathcal{I})$. If I_j is a proper interval

$[i, i+1]$, we introduce a variable $x_j \in \{s_{i,j}, \ldots, s_{i+1,j}\}$. Otherwise we do not need a variable x_j. The i-th constraint of $\mathrm{ILP}(\mathcal{I})$ algebraically represents $dist(\mathbf{x}, \mathbf{a}_i)$, see Example 6.

Observation 5. *The optimal value of* $\mathrm{ILP}(\mathcal{I})$ *is equal to* $\mathrm{OPT}(\mathcal{A}, \mathcal{I})$.

Example 6. Consider the following 5 sequences of length 7:

$$[a_{i,j}] = \begin{bmatrix} 20 & 18 & 20 & 10 & 16 & 8 & 10 \\ 11 & 6 & 7 & 17 & 14 & 14 & 17 \\ 14 & 12 & 18 & 13 & 11 & 6 & 12 \\ 19 & 8 & 16 & 18 & 12 & 19 & 19 \\ 16 & 15 & 11 & 15 & 6 & 17 & 11 \end{bmatrix}$$

and an interval system $\mathcal{I} = ([2,3],\ [2,3],\ [2,3],\ [3,4],\ [3,4],\ [3,3],\ [3,4])$. An illustration of both can be found in Fig. 2.

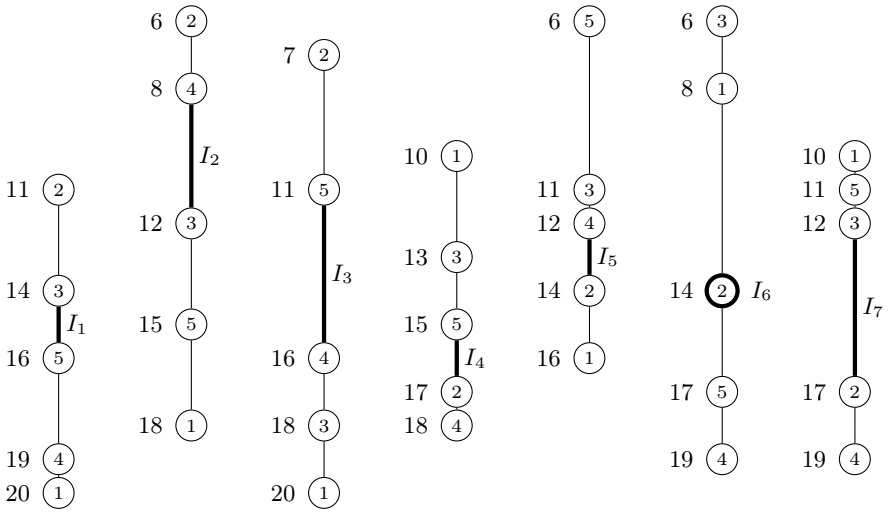

Fig. 2. Illustration of Example 6: 5 sequences of length 7 together with an interval system. Notice that I_6 is a degenerate interval.

We obtain the following $\mathrm{ILP}(\mathcal{I})$, where $x_1 \in [14, 16]$, $x_2 \in [8, 12]$, $x_3 \in [11, 16]$, $x_4 \in [15, 17]$, $x_5 \in [12, 14]$, $x_7 \in [12, 17]$ and the sequence $\mathbf{x}$ can be retrieved as $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, 14, x_7)$:

$$\begin{array}{llllllll} & & & & \min z & & & \\ 20 - x_1 & + 18 - x_2 & + 20 - x_3 & + x_4 - 10 & + 16 - x_5 & + 6 & + x_7 - 10 & \leq z \\ x_1 - 11 & + x_2 - 6 & + x_3 - 7 & + 17 - x_4 & + 14 - x_5 & + 0 & + 17 - x_7 & \leq z \\ x_1 - 14 & + 12 - x_2 & + 18 - x_3 & + x_4 - 13 & + x_5 - 11 & + 8 & + x_7 - 12 & \leq z \\ 19 - x_1 & + x_2 - 8 & + 16 - x_3 & + 18 - x_4 & + x_5 - 12 & + 5 & + 19 - x_7 & \leq z \\ 16 - x_1 & + 15 - x_2 & + x_3 - 11 & + x_4 - 15 & + x_5 - 6 & + 3 & + x_7 - 11 & \leq z \end{array}$$

Note that $P = \mathrm{ILP}(\mathcal{I})$ has the following special form, which we call (±)ILP:

$$\begin{aligned} &\min z \\ d_i + \sum_j x_j e_{i,j} &\le z \\ x_j &\in R_P(x_j) \end{aligned}$$

where $e_{i,j} = \pm 1$ and $R_P(x_j) = \{\ell_j, \ldots, r_j\}$ for integers $\ell_j \le r_j$. Whenever we refer to variables, it does not apply to z, which is of auxiliary character. Also, "$x_j \in R_P(x_j)$" are called variable ranges rather than constraints. We say that $(e_{1,j}, \ldots, e_{k,j})$ is a *coefficient vector* of x_j and denote it as $E_P(x_j)$. If the program P is apparent from the context, we omit the subscript.

Simplification of ILP. The following two facts are used to reduce the number of variables of a (±)ILP. For $A, B \subseteq \mathbb{Z}$ we define $-A = \{-a : a \in A\}$ and $A + B = \{a + b : a \in A, b \in B\}$.

Fact 7. *Let P be a* (±)ILP. *Let P' be a program obtained from P by replacing a variable x_j with $-x_j$, i.e. setting $E_{P'}(x_j) = -E_P(x_j)$ and $R_{P'}(x_j) = -R_P(x_j)$. Then* $\mathrm{OPT}(P) = \mathrm{OPT}(P')$.

Fact 8. *Let P be a* (±)ILP. *Assume $E_P(x_j) = E_P(x_{j'})$ for $j \neq j'$. Let P' be a program obtained from P by removing the variable $x_{j'}$ and replacing x_j with $x_j + x_{j'}$, i.e. setting $R_{P'}(x_j) = R_P(x_j) + R_P(x_{j'})$. Then* $\mathrm{OPT}(P) = \mathrm{OPT}(P')$.

Proof. Let $(z, x_1, \ldots, x_n)$ be a feasible solution of P. Then setting $x_j := x_j + x_{j'}$ and removing the variable $x_{j'}$ we obtain a feasible solution of P'. Therefore $\mathrm{OPT}(P') \le \mathrm{OPT}(P)$. For the proof of the other inequality, take a feasible solution $(z, x_1, \ldots, x_n)$ (with $x_{j'}$ missing) of P'. Note that $x_j \in R_{P'}(x_j) = R_P(x_j) + R_P(x_{j'})$. Therefore one can split x_j into $x_j + x_{j'}$ so that $x_j \in R_P(x_j)$ and $x_{j'} \in R_P(x_{j'})$. This way we obtain a feasible solution of P and thus prove that $\mathrm{OPT}(P) \le \mathrm{OPT}(P')$. □

Corollary 9. *For a* (±)ILP *with k constraints one can compute in linear time an equivalent* (±)ILP *with k constraints and up to 2^{k-1} variables.*

Proof. We apply Fact 7 to obtain $e_{1,1} = e_{1,2} = \ldots = e_{1,\ell}$, this leaves at most 2^{k-1} different coefficient vectors. Afterwards we apply Fact 8 as many times as possible until there is exactly one variable with each coefficient vector. □

Example 10. Consider the (±)ILP P from Example 6. Observe that $E_P(x_4) = E_P(x_7) = -E_P(x_2)$ and thus Facts 7 and 8 let us merge x_2 and x_7 into x_4 with

$$R_{P'}(x_4) = R_P(x_4) + R_P(x_7) - R_P(x_2) = [15, 17] + [12, 17] + [-12, -8] = [15, 26].$$

Simplifying the constant terms we obtain the following (±)ILP P':

$$\begin{array}{rl} & \min z \\ -x_1 - x_3 + x_4 - x_5 + 60 & \leq z \\ +x_1 + x_3 - x_4 - x_5 + 24 & \leq z \\ +x_1 - x_3 + x_4 + x_5 - 12 & \leq z \\ -x_1 - x_3 - x_4 + x_5 + 57 & \leq z \\ -x_1 + x_3 + x_4 + x_5 - 9 & \leq z \end{array}$$

3 Kernel of MSC for Arbitrary k

In this section we give a kernel for the MSC problem parameterized with k, which we then apply to develop a linear-time FPT algorithm. To obtain the kernel we need a combinatorial observation that if $\pi_j = \pi_{j'}$ then the j-th and the j'-th column in $\mathcal{A}$ can be merged. This is stated formally in the following lemma.

Lemma 11. *Let $\mathcal{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_k)$ be a collection of sequences of length ℓ. Assume that $\pi_j = \pi_{j'}$ for some $1 \leq j < j' \leq \ell$. Let $\mathcal{A}' = (\mathbf{a}'_1, \ldots, \mathbf{a}'_k)$ be a collection of sequences of length $\ell - 1$ obtained from $\mathcal{A}$ by removing the j'-th column and setting $a'_{i,j} = a_{i,j} + a_{i,j'}$. Then $\mathrm{OPT}(\mathcal{A}) = \mathrm{OPT}(\mathcal{A}')$.*

Proof. First, let us show that $\mathrm{OPT}(\mathcal{A}') \leq \mathrm{OPT}(\mathcal{A})$. Let $\mathbf{x}$ be a Manhattan consensus sequence for $\mathcal{A}$ and let $\mathbf{x}'$ be obtained from $\mathbf{x}$ by removing the j'-th entry and setting $x'_j = x_j + x_{j'}$. We claim that $\mathit{dist}(\mathbf{x}', \mathcal{A}') \leq \mathit{dist}(\mathbf{x}, \mathcal{A})$. Note that it suffices to show that $|x'_j - a'_{i,j}| \leq |x_j - a_{i,j}| + |x_{j'} - a_{i,j'}|$ for all i. However, with $x'_j = x_j + x_{j'}$ and $a'_{i,j} = a_{i,j} + a_{i,j'}$, this is a direct consequence of the triangle inequality.

It remains to prove that $\mathrm{OPT}(\mathcal{A}) \leq \mathrm{OPT}(\mathcal{A}')$. Let $\mathbf{x}'$ be a Manhattan consensus sequence for $\mathcal{A}'$. By Observation 4, x'_j is consistent with some proper basic interval $[i, i+1]$. Let $d'_{i,j} = x'_j - s'_{i,j}$ and $D'_{i,j} = s'_{i+1,j} - s'_{i,j}$. Also, let $D_{i,j} = s_{i+1,j} - s_{i,j}$ and $D_{i,j'} = s_{i+1,j'} - s_{i,j'}$. Note that, since $\pi_j = \pi_{j'}$, $D'_{i,j} = D_{i,j} + D_{i,j'}$. Thus, one can partition $d'_{i,j} = d_{i,j} + d_{i,j'}$ so that both $d_{i,j}$ and $d_{i,j'}$ are non-negative integers not exceeding $D_{i,j}$ and $D_{i,j'}$ respectively. We set $x_j = s_{i,j} + d_{i,j}$ and $x_{j'} = s_{i,j'} + d_{i,j'}$. The remaining components of $\mathbf{x}$ correspond to components of $\mathbf{x}'$. Note that $x'_j = x_j + x_{j'}$ and that both x_j and $x_{j'}$ are consistent with $[i, i+1]$. Consequently for any sequence $\mathbf{a}_m$ it holds that $\mathit{dist}(\mathbf{x}, \mathbf{a}_m) = \mathit{dist}(\mathbf{x}', \mathbf{a}_m)$ and therefore $\mathit{dist}(\mathbf{x}, \mathcal{A}) = \mathit{dist}(\mathbf{x}', \mathcal{A}')$, which concludes the proof.

Finally, note that the procedures given above can be used to efficiently convert between the optimum solutions $\mathbf{x}$ and $\mathbf{x}'$. □

By Lemma 11, to obtain the desired kernel we need to sort the elements in columns of $\mathcal{A}$ and afterwards group by the resulting permutations π_j.

Theorem 12. *In $\mathcal{O}(\ell k \log k)$ time one can reduce any instance of MSC to an instance with k sequences of length ℓ', with $\ell' \leq k!$.*

Remark 13. For binary instances, if permutations π_j are chosen appropriately, we can achieve $\ell' \leq 2^k$.

Theorem 14. *For any integer k, the* Manhattan Sequence Consensus *problem can be solved in $\mathcal{O}(\ell k \log k + 2^{k! \log k + O(k2^k)} \log M)$ time.*

Proof. We solve the kernel from Theorem 12 by considering all possible interval systems $\mathcal{I}$ composed of proper intervals. The sequences in the kernel have length at most $k!$, which gives $(k-1)^{k!}$ ($\pm$)ILPs of the form ILP($\mathcal{I}$) to solve.

Each of the ($\pm$)ILPs initially has k constraints on $k!$ variables but, due to Corollary 9, the number of variables can be reduced to 2^{k-1}. Lenstra's algorithm with further improvements [13,9,18] solves ILP with p variables in $\mathcal{O}(p^{2.5p+o(p)} \log L)$ time, where L is the bound on the scope of variables. In our case $L = \mathcal{O}(\ell M)$, which gives the time complexity of:

$$\mathcal{O}\left((k-1)^{k!} \cdot 2^{(k-1)(2.5 \cdot 2^{k-1} + o(2^{k-1}))} \log M\right) = \mathcal{O}\left(2^{k! \log k + O(k2^k)} \log M\right).$$

This concludes the proof of the theorem. □

4 Combinatorial Characterization of Solutions for $k = 5$

In this section we characterize those Manhattan consensus sequences $\mathbf{x}$ which additionally minimize $\sum_i dist(\mathbf{x}, \mathbf{a}_i)$ among all Manhattan consensus sequences. Such sequences are called here *sum-MSC sequences*. We show that one can determine a collection of 20 interval systems, so that any sum-MSC sequence is guaranteed to be consistent with one of them. We also prove some structural properties of these systems, which are then useful to efficiently solve the corresponding ($\pm$)ILPs.

We say that x_j is in the *center* if $x_j = s_{3,j}$, i.e. x_j is equal to the column median. Note that if $x_j \neq s_{3,j}$, then moving x_j by one towards the center decreases by one $dist(\mathbf{x}, \mathbf{a}_i)$ for at least three sequences $\mathbf{a}_i$.

Definition 15. *We say that $\mathbf{a}_i$* governs *x_j if x_j is in the center or moving x_j towards the center increases $dist(\mathbf{x}, \mathbf{a}_i)$. The set of indices i such that $\mathbf{a}_i$ governs x_j is denoted as $G_j(\mathbf{x})$.*

Observe that if x_j is in the center, then $|G_j(\mathbf{x})| = 5$, and otherwise $|G_j(\mathbf{x})| \leq 2$; see Fig. 3. For $k = 5$ we have 4 proper basic intervals: $[1,2]$, $[2,3]$, $[3,4]$ and $[4,5]$. We call $[1,2]$ and $[4,5]$ *border* intervals, and the other two *middle intervals*. We define $G_j([1,2]) = \{\pi_j(1)\}$, $G_j([2,3]) = \{\pi_j(1), \pi_j(2)\}$, $G_j([3,4]) = \{\pi_j(4), \pi_j(5)\}$ and $G_j([4,5]) = \{\pi_j(5)\}$. Note that if we know $G_j(\mathbf{x})$ and $|G_j(\mathbf{x})| \leq 2$, then we are guaranteed that x_j is consistent with the basic interval I_j for which $G_j(I_j) = G_j(\mathbf{x})$.

Observe that if $\mathbf{x}$ is a Manhattan consensus sequence, then $G_j(\mathbf{x}) \neq \emptyset$ for any j. If we additionally assume that $\mathbf{x}$ is a sum-MSC sequence, we obtain a stronger property.

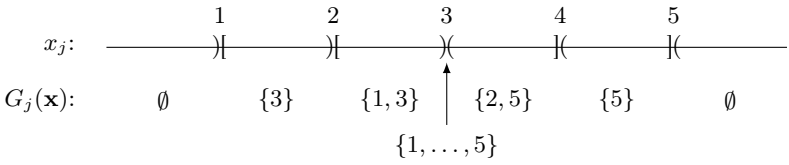

Fig. 3. Assume $a_{1,j} = 2, a_{2,j} = 4, a_{3,j} = 1, a_{4,j} = 3, a_{5,j} = 5$. Then $G_j(\mathbf{x})$ depends on the interval of x_j as shown in the figure; e.g., if $1 \leq x_j < 2$ then $G_j(\mathbf{x}) = \{3\}$.

Lemma 16. *Let* $\mathbf{x}$ *be a sum-MSC sequence. Then* $G_j(\mathbf{x}) \cap G_{j'}(\mathbf{x}) \neq \emptyset$ *for any* j, j'.

Proof. For a proof by contradiction assume $G_j(\mathbf{x})$ and $G_{j'}(\mathbf{x})$ are disjoint. This implies that neither x_j nor $x_{j'}$ is in the center and thus $|G_j(\mathbf{x})|, |G_{j'}(\mathbf{x})| \leq 2$. Let us move both x_j and $x_{j'}$ by one towards the center. Then $dist(\mathbf{x}, \mathbf{a}_i)$ remains unchanged for $i \in G_j(\mathbf{x}) \cup G_{j'}(\mathbf{x})$ (by disjointness), and decreases by two for the remaining sequences $\mathbf{a}_i$. There must be at least one such remaining sequence, which contradicts our choice of $\mathbf{x}$. □

Additionally, if a sum-MSC sequence $\mathbf{x}$ has a position j with $|G_j(\mathbf{x})| = 1$, the structure of $\mathbf{x}$ needs to be even more regular.

Definition 17. *A sequence* $\mathbf{x}$ *is called an* i-border sequence *if for each* j *it holds that* $x_j = a_{i,j}$ *or* $G_j(\mathbf{x}) = \{i\}$.

Lemma 18. *Let* $\mathbf{x}$ *be a sum-MSC sequence. If* $G_j(\mathbf{x}) = \{i\}$ *for some* j, *then* $\mathbf{x}$ *is an* i*-border sequence.*

Proof. For a proof by contradiction assume $G_{j'}(\mathbf{x}) \neq \{i\}$ and $x_{j'} \neq a_{i,j'}$ for some j'. Let us move x_j towards the center and $x_{j'}$ towards $a_{i,j'}$ both by one. Then for any i' it holds that $dist(\mathbf{x}, \mathbf{a}_{i'})$ does not increase. By Lemma 16 $i \in G_{j'}(\mathbf{x})$, so $x_{j'}$ is moved away from the center. Moreover, x_j is moved towards some $a_{i',j}$ with $i' \neq i$, since $G_{j'}(\mathbf{x}) \neq \{i\}$. Consequently, $dist(\mathbf{x}, \mathbf{a}_{i'})$ decreases by two, which contradicts our choice of $\mathbf{x}$. □

Definition 19. *A sequence* $\mathbf{x}$ *is called an* i-middle sequence *if for each* j *it holds that* $i \in G_j(\mathbf{x})$ *and* $|G_j(\mathbf{x})| \geq 2$.

Definition 20. *For a 3-element set* $\Delta \subseteq \{1, \ldots, 5\}$ *a sequence* $\mathbf{x}$ *is called a* Δ-triangle sequence *if for each* j *it holds that* $|\Delta \cap G_j(\mathbf{x})| \geq 2$.

Lemma 21. *Let* $\mathbf{x}$ *be a sum-MSC sequence. Then* $\mathbf{x}$ *is a border sequence, a middle sequence or a triangle sequence.*

Proof. Recall that $G_j(\mathbf{x}) \neq \emptyset$ for each j. By Lemma 18, if $G_j(\mathbf{x}) = \{i\}$ for some j, then $\mathbf{x}$ is an i-border sequence. This lets us assume $|G_j(x)| \geq 2$, i.e. $|G_j(\mathbf{x})| \in \{2, 5\}$, for each j. Let $\mathcal{F}$ be the family of 2-element sets among $G_j(\mathbf{x})$. By Lemma 16 every two of them intersect, so we can apply the following easy set-theoretical claim.

Claim. Let $\mathcal{G}$ be a family of 2-element sets such that every two sets in $\mathcal{G}$ intersect. Then sets in $\mathcal{G}$ share a common element or $\mathcal{G}$ contains exactly three sets with three elements in total.

If all sets in $\mathcal{F}$ share an element i, then $\mathbf{x}$ is clearly an i-middle sequence. Otherwise $\mathbf{x}$ is a Δ-triangle sequence for $\Delta = \bigcup \mathcal{F}$. □

Fact 22. *There exist 20 interval systems $\mathcal{B}_i, \mathcal{M}_i$ (for $i \in \{1, \ldots, 5\}$) and $\mathcal{T}_\Delta$ (for 3-element sets $\Delta \subseteq \{1, \ldots, 5\}$) such that:*
(a) $\mathcal{B}_i$ is consistent with all i-border sequences, $G_j(\mathcal{B}_{i,j}) = \{i\}$ for proper $\mathcal{B}_{i,j}$;
(b) $\mathcal{M}_i$ is consistent with all i-middle sequences and if $\mathcal{M}_{i,j}$ is proper then $|G_j(\mathcal{M}_{i,j})| = 2$ and $i \in G_j(\mathcal{M}_{i,j})$;
(c) $\mathcal{T}_\Delta$ is consistent with all Δ-triangle sequences and if $\mathcal{T}_{\Delta,j}$ is proper then $|G_j(\mathcal{T}_{\Delta,j})| = 2$ and $G_j(\mathcal{T}_{\Delta,j}) \subseteq \Delta$.

Proof. (a) Let us fix a position j. For any i-border sequence $\mathbf{x}$, we know that $x_j = a_{i,j}$ or $G_j(\mathbf{x}) = \{i\}$. If either of the border intervals I satisfies $G_j(I) = \{i\}$, we set $\mathcal{B}_{i,j} := I$ (observe that $a_{i,j}$ is then consistent with I). Otherwise we choose $\mathcal{B}_{i,j}$ so that it is degenerate and corresponds to $x_j = a_{i,j}$.
(b) Again fix j. For any i-middle sequence $\mathbf{x}$, we know that x_j is consistent with at least one of the two middle intervals (both if x_j is in the center). If either of the middle intervals I satisfies $i \in G_j(I)$, we choose $\mathcal{M}_{i,j} := I$. (Note that this condition cannot hold for both middle intervals). Otherwise we know that x_j is in the center and set $\mathcal{M}_{i,j}$ so that it is degenerate and corresponds to x_j in the center, i.e. $\mathcal{M}_{i,j} := [3, 3]$.
(c) We act as in (b), i.e. if either of the middle intervals I satisfies $|G_j(I) \cap \Delta| = 2$, we choose $\mathcal{T}_{\Delta,j} := I$ (because sets $G_j(I)$ are disjoint for both middle intervals, this condition cannot hold for both of them). Otherwise, we set $\mathcal{T}_{\Delta,j} := [3, 3]$, since x_j is guaranteed to be in the center for any Δ-triangle sequence $\mathbf{x}$. □

5 Practical Algorithm for $k \leq 5$

It suffices to consider $k = 5$. Using Fact 22 we reduce the number of interval systems from $(k-1)^{k!} = 4^{5!} > 10^{72}$ to 20 compared to the algorithm of Section 3. Moreover, for each of them ILP($\mathcal{I}$) admits structural properties, which lets us compute OPT($\mathcal{A}, \mathcal{I}$) much more efficiently than using a general ILP solver.

Definition 23. *A* (±)ILP *is called* easy *if for each constraint the number of* +1 *coefficients is* 0, 1 *or* n, *where* n *is the number of variables.*

Lemma 24. *For each $\mathcal{I}$ being one of the 20 interval systems $\mathcal{B}_i, \mathcal{M}_i$ and $\mathcal{T}_\Delta$,* ILP($\mathcal{I}$) *can be reduced to an equivalent easy* (±)ILP *with up to 4 variables.*

Proof. Recall that for degenerate intervals I_j, we do not introduce variables. On the other hand, if I_j is proper, possibly negating the variable x_j (Fact 7), we can make sure that the coefficient vector $E(x_j)$ has +1 entries corresponding to $i \in G_j(I_j)$ and −1 entries for the remaining i. Moreover, merging the variables

(Fact 8), we end up with a single variable per possible value $G_j(I_j)$. Now we use structural properties stated in Fact 22 to claim that the (±)ILP we obtain this way, possibly after further variable negations, becomes easy.

Border Sequences. By Fact 22(a), if $\mathcal{B}_{i,j}$ is proper, then $G_j(\mathcal{B}_{i,j}) = \{i\}$ and thus the (±)ILP has at most 1 variable and consequently is easy.

Middle Sequences. By Fact 22(b), if $\mathcal{M}_{i,j}$ is proper, then $G_j(\mathcal{M}_{i,j}) = \{i, i'\}$ for some $i' \neq i$. Thus there are up to 4 variables, the constraint corresponding to i has only $+1$ coefficients, and the remaining constraints have at most one $+1$.

Triangle Sequences. By Fact 22(c), if $\mathcal{T}_{\Delta,j}$ is proper, then $G_j(\mathcal{T}_{\Delta,j})$ is a 2-element subset of Δ, and thus there are up to three variables. Any (±)ILP with up to two variables is easy, and if we obtain three variables, then the constraints corresponding to $i \in \Delta$ have exactly two $+1$ coefficients, while the constraints corresponding to $i \notin \Delta$ have just -1 coefficients. Now, negating each variable (Fact 7), we get one $+1$ coefficient in constraints corresponding to $i \in \Delta$ and all $+1$ coefficients for $i \notin \Delta$. □

The algorithm of Lenstra [17] with further improvements [13,9,18], which runs in roughly $n^{2.5n+o(n)}$ time, could perform reasonably well for $n = 4$. However, there is a simple $\mathcal{O}(n^2)$-time algorithm designed for easy (±)ILP. Due to space constraints, it is omitted in this paper. It can be found in the full version [14].

In conclusion, the algorithm for MSC problem first proceeds as described in Fact 22 to obtain the interval systems $\mathcal{B}_i, \mathcal{M}_i$ and $\mathcal{T}_\Delta$. For each of them it computes ILP($\mathcal{I}$), as described in Section 2, and converts it to an equivalent easy (±)ILP following Lemma 24. Finally, it uses the efficient algorithm to solve each of these 20 (±)ILPs. The final result is the minimum of the optima obtained.

6 Conclusions

We have presented an $\mathcal{O}(\ell k \log k)$-time kernelization algorithm, which for any instance of the MSC problem computes an equivalent instance with $\ell' \leq k!$. Although for $k \leq 5$ this gives an instance with $\ell' \leq 120$, i.e. the kernel size is constant, solving it in a practically feasible time remains challenging. Therefore for $k \leq 5$ we have designed an efficient linear-time algorithm.

We have implemented the algorithm,[1] including retrieving the optimum consensus sequence (omitted in the description above). For random input data with $\ell = 10^6$ and $k = 5$, the algorithm without kernelization achieved the running time of 1.48015s, which is roughly twice the time required to read the input file (0.73443s, not included in the former). The algorithm pipelined with the kernelization achieved 0.33415s. The experiments were conducted on a MacBook Pro notebook (2.3 Ghz Intel Core i7, 8 GB RAM).

References

1. Amir, A., Paryenty, H., Roditty, L.: Configurations and minority in the string consensus problem. In: Calderón-Benavides, L., González-Caro, C., Chávez, E., Ziviani, N. (eds.) SPIRE 2012. LNCS, vol. 7608, pp. 42–53. Springer, Heidelberg (2012)

[1] Source code is available at `http://www.mimuw.edu.pl/~kociumaka/files/msc.cpp`.

2. Amir, A., Paryenty, H., Roditty, L.: On the hardness of the consensus string problem. Inf. Process. Lett. 113(10-11), 371–374 (2013)
3. Andoni, A., Indyk, P., Patrascu, M.: On the optimality of the dimensionality reduction method. In: FOCS, pp. 449–458. IEEE Computer Society (2006)
4. Badoiu, M., Har-Peled, S., Indyk, P.: Approximate clustering via core-sets. In: Reif, J.H. (ed.) STOC, pp. 250–257. ACM (2002)
5. Boucher, C., Brown, D.G., Durocher, S.: On the structure of small motif recognition instances. In: Amir, A., Turpin, A., Moffat, A. (eds.) SPIRE 2008. LNCS, vol. 5280, pp. 269–281. Springer, Heidelberg (2008)
6. Cohen, G.D., Honkala, I.S., Litsyn, S., Solé, P.: Long packing and covering codes. IEEE Transactions on Information Theory 43(5), 1617–1619 (1997)
7. Fischer, K., Gärtner, B., Kutz, M.: Fast smallest-enclosing-ball computation in high dimensions. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 630–641. Springer, Heidelberg (2003)
8. Frances, M., Litman, A.: On covering problems of codes. Theory Comput. Syst. 30(2), 113–119 (1997)
9. Frank, A., Tardos, É.: An application of simultaneous diophantine approximation in combinatorial optimization. Combinatorica 7(1), 49–65 (1987)
10. Gärtner, B., Schönherr, S.: An efficient, exact, and generic quadratic programming solver for geometric optimization. In: Symposium on Computational Geometry, pp. 110–118 (2000)
11. Graham, R.L., Sloane, N.J.A.: On the covering radius of codes. IEEE Transactions on Information Theory 31(3), 385–401 (1985)
12. Gramm, J., Niedermeier, R., Rossmanith, P.: Fixed-parameter algorithms for closest string and related problems. Algorithmica 37(1), 25–42 (2003)
13. Kannan, R.: Minkowski's convex body theorem and integer programming. Mathematics of Operations Reasearch 12, 415–440 (1987)
14. Kociumaka, T., Pachocki, J.W., Radoszewski, J., Rytter, W., Waleń, T.: On the string consensus problem and the Manhattan sequence consensus problem (full version). CoRR, abs/1407.6144 (2014)
15. Kumar, P., Mitchell, J.S.B., Yildirim, E.A.: Computing core-sets and approximate smallest enclosing hyperspheres in high dimensions. In: 5th Workshop on Algorithm Engineering and Experiments (2003)
16. Lanctôt, J.K., Li, M., Ma, B., Wang, S., Zhang, L.: Distinguishing string selection problems. In: Tarjan, R.E., Warnow, T. (eds.) SODA, pp. 633–642. ACM/SIAM (1999)
17. Lenstra Jr., H.W.: Integer programming with a fixed number of variables. Mathematics of Operations Research 8, 538–548 (1983)
18. Lokshtanov, D.: New Methods in Parameterized Algorithms and Complexity. PhD thesis, University of Bergen (2009)
19. Ma, B., Sun, X.: More efficient algorithms for closest string and substring problems. SIAM J. Comput. 39(4), 1432–1443 (2009)
20. Mazumdar, A., Polyanskiy, Y., Saha, B.: On Chebyshev radius of a set in Hamming space and the closest string problem. In: ISIT, pp. 1401–1405. IEEE (2013)
21. Ritter, J.: An efficient bounding sphere. In: Glassner, A.S. (ed.) Gems. Academic Press, Boston (1990)
22. Sylvester, J.J.: A question in the geometry of situation. Quarterly Journal of Pure and Applied Mathematics 1, 79 (1857)