Improved Lower Bounds for the Shortest Superstring and Related Problems

Marek Karpinski* Richard Schmied†

Abstract

We study the approximation hardness of the Shortest Superstring, the Maximal Compression and the Maximum Asymmetric Traveling Salesperson (MAX-ATSP) problem. We introduce a new reduction method that produces strongly restricted instances of the Shortest Superstring problem, in which the maximal orbit size is eight (with no character appearing more than eight times) and all given strings having length at most six. Based on this reduction method, we are able to improve the best up to now known approximation lower bound for the Shortest Superstring problem and the Maximal Compression problem by an order of magnitude. The results imply also an improved approximation lower bound for the MAX-ATSP problem.

1 Introduction

In the **Shortest Superstring problem**, we are given a finite set S of strings and we would like to construct their shortest superstring, which is the shortest possible string such that every string in S is a proper substring of it.

The task of computing a shortest common superstring appears in a wide variety of application related to computational biology (see. e.g. [L88] and [L90]). Intuitively, short superstrings preserve important biological structure and are good models of the original DNA sequence. In context of computational biology, DNA sequencing is the important task of determining the sequence of nucleotides in a molecule of DNA. The DNA can be seen as a double-stranded sequence of four types of nucleotides represented by the alphabet $\{a, c, g, t\}$. Identifying those strings for different molecules is an important step towards understanding the biological functions of the molecules. However, with current laboratory methods, it is quite impossible to extract a long molecule directly as a whole. In fact, biochemists split millions of identical molecules into pieces each typically containing

*Dept. of Computer Science and the Hausdorff Center for Mathematics, University of Bonn. Supported in part by DFG grants and the Hausdorff Center grant EXC59-1. Email: marek@cs.uni-bonn.de

†Dept. of Computer Science, University of Bonn. Work supported by Hausdorff Doctoral Fellowship. Email: schmied@cs.uni-bonn.de

at most 500 nucleotides. Then, from sometimes millions of these fragments, one has to compute the superstring representing the whole molecule.

From the computational point of view, the Shortest Superstring problem is an optimization problem, which consists of finding a minimum length superstring for a given set S of strings over a finite alphabet Σ. The underlying decision version was proved to be $\mathbf{NP}$-complete [MS77]. However, there are many applications that involve relatively simple classes of strings. Motivated by those applications, many authors have investigated whether the Shortest Superstring problem becomes polynomial time solvable under various restrictions to the set of instances. Gallant et al. [GMS80] proved that this problem in the exact setting is still $\mathbf{NP}$-complete for strings of length three and polynomial time solvable for strings of length two. On the other hand, Timkovskii [T90] studied the Shortest Superstring problem under restrictions to the orbit size of the letters in Σ. The orbit size of a letter is the number of its occurrences in the strings of S. Timkovskii proved that this problem restricted to instances with maximal orbit size two is polynomial time solvable. He raised the question about the status of the problem with maximal orbit size k for any constant $k \geq 3$. It is known that the Shortest Superstring problem remains $\mathbf{NP}$-hard for the following strongly restricted instances, such as

(i) all strings have length four and the maximal orbit size is six [M94],

(ii) the size of the alphabet of the instance is exactly two [GMS80], and

(iii) all strings are of the form $10p10q$, whereby $p, q \in \mathbb{N}$ [M98].

In order to cope with the exact computation intractability, approximation algorithms were designed to deal with this problem. The first polynomial time approximation algorithm with a constant approximation ratio was given by Blum et al. [BJL+94]. It achieves an approximation ratio 3. This factor was improved in a series of papers yielding approximation ratios of 2.88 by Teng and Yao [TT93]; 2.83 by Czumaj et al. [CGP+94]; 2.79 by Kosaraju, Park, and Stein [KPS94]; 2.75 by Armen and Stein [AS95]; 2.67 by Armen and Stein [AS98] and 2.596 by Breslauer, Jiang, and Jiang [BJJ97]. The currently best known approximation algorithm is due to Sweedyk [S99] and yields an approximation ratio of 2.5.

On the lower bound side, Blum et al. [BJL+94] proved that approximating the Shortest Superstring problem is $\mathbf{APX}$-hard. However, the constructed reduction produces instances with arbitrarily large alphabets. In [O99], Ott provided the first explicit approximation hardness result and proved that the problem is $\mathbf{APX}$-hard even if the size of the alphabet is two. In fact, Ott proved that instances over a binary alphabet are $\mathbf{NP}$-hard to approximate with an approximation ratio $\frac{17246}{17245}$ $(1.000057) - \epsilon$ for every $\epsilon > 0$. In 2005, Vassilevska [V05] established an improved approximation lower bound of $\frac{1217}{1216}$ (1.00082) by using a natural construction. The instances constructed in [V05] have a special structure. More precisely, Vassilevska proved that there are instances of the Shortest Superstring problem with maximal orbit size 20 such that approximating these instances with less than $\frac{1217}{1216}$ (1.00082) is $\mathbf{NP}$-hard. In this paper, we prove that even instances of the Shortest Superstring problem with maximal orbit size 8 are $\mathbf{NP}$-hard to approximate with less than $\frac{345}{344}$ (1.0029).

Maximal Compression problem. We are given a collection of strings $S = \{s_1, \ldots, s_n\}$. The task is to find a superstring for S with maximum compression, which is the difference between the sum of the lengths of the given strings and the length of the superstring.

In the exact setting, an optimal solution to the Shortest Superstring problem is an optimal solution to this problem, but the approximate solutions can differ significantly in the sense of approximation ratio. The Maximal Compression problem arises in various data compression problems (cf. [SS82], [S88] and [MJ75]). The decision version of this problem is $\mathbf{NP}$-complete [MS77]. Tarhio and Ukkonen [TU88] and Turner [T89] gave approximation algorithms with approximation ratio 2. The best known approximation upper bound is 1.5 [KLS$^+$05] by reducing it to the MAX-ATSP problem, which is defined below.

On the approximation lower bound side, Blum et al. [BJL$^+$94] proved the $\mathbf{APX}$-hardness of the Maximal Compression problem. The first explicit approximation lower bounds were given by Ott [O99], who proved that it is $\mathbf{NP}$-hard to approximate this problem with an approximation factor $\frac{11217}{11216}$ $(1.000089) - \epsilon$ for every $\epsilon > 0$. This hardness result was improved by Vassilevska [V05] implying a lower bound of $\frac{1072}{1071}$ $(1.00093) - \epsilon$ for any $\epsilon > 0$, unless $\mathbf{P} = \mathbf{NP}$. In this paper, we prove that approximating the Maximal Compression problem with an approximation ratio less than $\frac{207}{206}$ (1.0048) is $\mathbf{NP}$-hard.

Maximum Asymmetric Traveling Salesperson (MAX-ATSP) problem. We are given a complete directed graph G and a weight function w assigning each edge of G a nonnegative weight. The task is to find a closed tour of maximum weight visiting every vertex of G exactly once .

This problem has various applications and in fact, a good approximation algorithm for MAX-ATSP yields a good approximation algorithm for many other optimization problems such as the Shortest Superstring problem, the Maximum Compression problem and the Minimum Asymmetric $(1,2)$-Traveling Salesperson (MIN-$(1,2)$-ATSP) problem. The latter problem is the restricted version of the Minimum Asymmetric Traveling Salesperson problem, in which we restrict the weight function w to weights one and two. The MAX-ATSP problem can be seen as a generalization of the MIN-$(1,2)$-ATSP problem in the sense that any $(\frac{1}{\alpha})$-approximation algorithm for the former problem transforms in a $(2-\alpha)$- approximation algorithm for the latter problem. Due to this reduction, all negative results concerning the approximation of the MIN-$(1,2)$-ATSP problem imply hardness results for the MAX-ATSP problem. Since MIN-$(1,2)$-ATSP is $\mathbf{APX}$-hard [PY93], there is little hope for polynomial time approximation algorithms with arbitrary good precision for the MAX-ATSP problem. On the other hand, the first approximation algorithm for the MAX-ATSP problem with guaranteed approximation performance is due to Fisher, Nemhauser, and Wolsey [FNW79] and achieves an approximation factor of 2. After that Kosaraju, Park, and Stein [KPS94] gave an approximation algorithm for that problem with performance ratio 1.66. This result was improved by

Bläser [B02] who obtained an approximation upper bound of 1.63. Lewenstein and Sviridenko [LS03] were able to improve the approximation upper bound for that problem to 1.60. Then, Kaplan et al. [KLS$^+$05] designed an algorithm for the MAX-ATSP problem yielding the best known approximation upper bound of 1.50.

On the approximation hardness side, Engebretsen [E99] proved that, for any $\epsilon > 0$, there is no $(\frac{2805}{2804} - \epsilon)$-approximation algorithm for MIN-$(1,2)$-ATSP, unless $\mathbf{P} = \mathbf{NP}$, which yields an approximation lower bound of $\frac{2804}{2803}$ $(1.00035) - \epsilon$ for the MAX-ATSP problem. The negative result was improved by Engebretsen and Karpinski [EK06] to $\frac{321}{320}$ $(1.0031) - \epsilon$ for the MIN-$(1,2)$-ATSP problem. It implies the best known approximation lower bound of $\frac{320}{319}$ $(1.0031) - \epsilon$, unless $\mathbf{P} = \mathbf{NP}$. In this paper, we prove that approximating the MAX-ATSP problem with an approximation ratio less than $\frac{207}{206}$ (1.0048) is $\mathbf{NP}$-hard.

2 Preliminaries

In the following, we introduce some notation and abbreviations.

Throughout, for $i \in \mathbb{N}$, we use the abbreviation $[i]$ for the set $\{1, \ldots, i\}$. Given an finite alphabet Σ, a string is an element of Σ^*. Given two strings $v = v_1 \cdots v_n$ and $w = w_1 \cdots w_m$ over Σ, we define the length of v denoted by $|v|$ as n. Furthermore, v is a substring of w, if $m \geq n$ and there exists a $j \in \{0, .., n-m\}$ such that for all $i \in [m]$, $v_i = w_{j+i}$. w is said to be a superstring of v if v is a substring of w. Given a set of strings $S = \{s_1, ..., s_n\} \subset \Sigma^*$, a string $s \in \Sigma^*$ is a superstring for S if s is a superstring of every $s_i \in S$. Given a superstring s for S, the compression of s, denoted as $comp(S, s)$ is defined as

$$comp(S, s) = \sum_{s_i \in S} |s_i| - |s|.$$

In addition, we introduce the notion of the maximal orbit size of S which is given by

$$\max_{a \in \Sigma} |\{s_i^j \in \Sigma \mid a = s_i^j, s_i = s_i^1 \cdots s_i^{n_i}, s_i \in S\}|.$$

We are ready to give the definition of the Shortest Superstring problem and the Maximal Compression problem.

Definition 1. *Given an alphabet Σ and a set of strings $S = \{s_1, ..., s_n\} \subset \Sigma^*$ such that no string in S is a substring of another string in S, in the Shortest Superstring problem we have to find a string s for S of minimum length, whereas in the Maximum Compression problem, we have to find a superstring s for S with maximum compression.*

In the following, we concentrate on the traveling salesperson problems. We begin with the definition of the MAX-ATSP problem. For this reason, we introduce the notion of a *Hamiltonian tour*. Given a directed graph $G = (V, A)$, a Hamiltonian tour is a cycle in G visiting each vertex of G exactly once.

Definition 2 (MAX-ATSP)**.** *Given a complete directed graph $G = (V, A)$ and a weight function w assigning each edge of G a nonnegative weight, the MAX-ATSP problem consists of finding a Hamiltonian tour of maximum weight in G.*

Next, we give the definition of the MIN-$(1, 2)$-ATSP problem, which is closely related to the MAX-ATSP problem.

Definition 3 (MIN-$(1, 2)$-ATSP)**.** *In the MIN-$(1, 2)$-ATSP problem, we are given a complete directed graph $G = (V, A)$ and a weight function $w : A \to \{1, 2\}$. The task is to find a Hamiltonian tour of minimum weight in G.*

3 Related Work

In the following, we present some results related to the problems studied in this paper. In particular, we describe briefly some reductions, which we use later on.

The following theorem is due to Vassilevska [V05] and deals with best known approximation lower bounds for the Shortest Superstring problem as well as for the Maximal Compression problem.

Theorem 1 ([V05])**.** *For any $\epsilon > 0$, it is $\mathbf{NP}$-hard to approximate the Shortest Superstring problem and the Maximal Compression problem restricted to instances with equal length strings in polynomial time within a factor of*

- $1.00082 - \epsilon$ *and*
- $1.00093 - \epsilon$*, respectively.*

In addition, the maximal orbit size of the constructed instances in [V05] is 20 and all strings have length four. In the same paper, it was proved that the Shortest Superstring problem is the hardest to approximate on instances over a binary alphabet.

Theorem 2 ([V05])**.** *Suppose the Shortest Superstring problem can be approximated by a factor α on instances over a binary alphabet. Then, the Shortest Superstring problem can be approximated by a factor α on instances over any alphabet.*

Given an instance S of the Shortest Superstring problem, consider the associated weighted complete graph, in which the vertices are represented by the strings in S and the weight of an edge is given by the the number of maximum overlapped letters of the corresponding strings. Then, the optimal compression is equivalent to the weight of a maximum Hamiltonian path. By introducing a special vertex representing the start and the end of the Hamiltonian cycle, the Maximal Compression problem is equivalent to the MAX-ATSP problem on this graph. This fact was used in [KLS$^+$05] in order to obtain an improved approximation algorithm for the Maximal Compression problem.

Fact 1. *An α-approximation algorithm for the MAX-ATSP problem implies an α-approximation algorithm for the Maximal Compression problem.*

Another interesting relation can be derived by replacing all edges with weight two of an instance of the MIN-$(1, 2)$-ATSP problem by edges of weight zero and then, computing a Hamiltonian tour of maximum weight. Vishwanathan[V92] proved that this transformation relates the MIN-$(1, 2)$-ATSP problem to the MAX-ATSP problem in the following sense.

Theorem 3 ([V92])**.** *An $(\frac{1}{\alpha})$- approximation algorithm for the MAX-ATSP problem implies an $(2 - \alpha)$- approximation algorithm for the MIN-$(1, 2)$-ATSP problem.*

Due to this reduction, every hardness result concerning the MIN-$(1, 2)$-ATSP problem can be transformed into a hardness result for the MAX-ATSP problem. The best known approximation lower bound for the MIN-$(1, 2)$-ATSP problem is proved in [EK06] and it yields the following hardness result .

Theorem 4 ([EK06])**.** *For any constant $\epsilon > 0$, it is $\mathbf{NP}$-hard to approximate the MIN-$(1, 2)$-ATSP problem with an approximation ratio $1.0031 - \epsilon$.*

According to Theorem 3, it implies the hardness result for the MAX-ATSP problem stated below.

Corollary 1. *For any constant $\epsilon > 0$, it is $\mathbf{NP}$-hard to approximate the MAX-ATSP problem within $1.0031 - \epsilon$.*

3.1 Hybrid Problem

In their paper on approximation hardness of bounded occurrence instances of several combinatorial optimization problems, Berman and Karpinski [BK99] introduced the Hybrid problem and proved that this problem is $\mathbf{NP}$-hard to approximate with some constant.

Definition 4 (Hybrid problem)**.** *Given a system of linear equations mod 2 containing n variables, m_2 equations with exactly two variables, and m_3 equations with exactly three variables, find an assignment to the variables that satisfies as many equations as possible.*

In the aforementioned paper, Berman and Karpinski proved the following hardness result.

Theorem 5 ([BK99])**.** *For any constant $\epsilon > 0$, there exists instances of the Hybrid problem with 42ν variables, 60ν equations with exactly two variables, and 2ν equations with exactly three variables such that:*

(i) *Each variable occurs exactly three times.*

(ii) *Either there is an assignment to the variables that leaves at most $\epsilon\nu$ equations unsatisfied, or else every assignment to the variables leaves at least $(1 - \epsilon)\nu$ equations unsatisfied.*

(iii) *It is* $\mathbf{NP}$*-hard to decide which of the two cases in item* (ii) *above holds.*

Analyzing the details of their construction, it can be seen that every instance of the Hybrid problem produced by it has an even more special structure. The equations containing three variables are of the form $x \oplus y \oplus z = \{0, 1\}$. Those equations arise from the Theorem of Håstad [H01] concerning the hardness of approximating equations with exactly three variables called the MAX-E3-LIN problem, which can be seen as a special instance of the Hybrid problem.

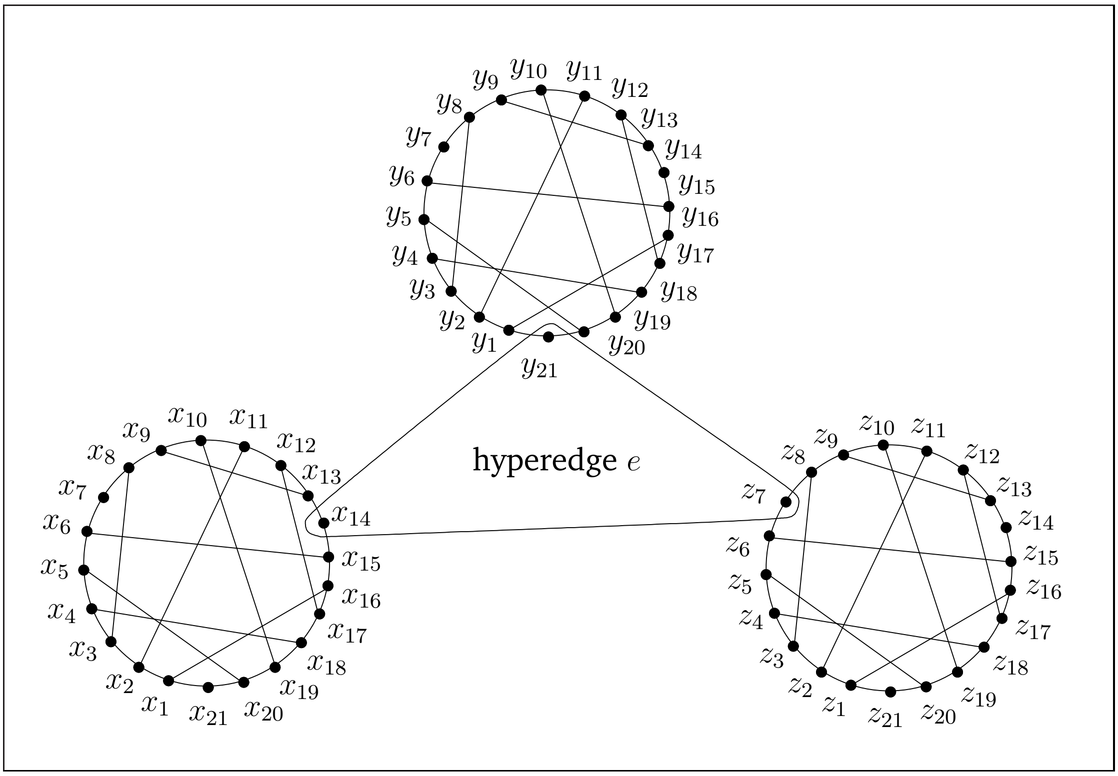

Figure 1: An example of a Hybrid instance with circles C^x, C^y, C^z, and hyperedge $e = \{z_7, y_{21}, x_{14}\}$.

For every variable x of the original instance $\mathcal{E}_3$ of the MAX-E3-LIN problem, they introduced a corresponding set of variables V_x. If the variable x occurs t_x times in $\mathcal{E}_3$, then, V_x contains $7t_x$ variables $x_1, \dots, x_{7t_x}$. Furthermore, the variables in V_x are connected by equations of the form $x_i \oplus x_{i+1} = 0$ with $i \in [7t_x - 1]$ and $x_1 \oplus x_{7t_x} = 0$. This construction induces the circle C^x on the variables V_x. In addition to it, every circle C^x possesses an associated matching M^x. The variables contained in $CV_x = \{x_i \mid i \in \{7\nu \mid \nu \in [t_x]\}\}$ are called *contact variables*, whereas the variables in $V_x \backslash CV_x$ are called *checker variables*.

Let $\mathcal{E}_3$ be an instance of the MAX-E3-LIN problem and $\mathcal{H}$ be its corresponding instance of the Hybrid problem. We denote by $V(\mathcal{E}_3)$ the set of variables which occur in the instance $\mathcal{E}_3$. Then, $\mathcal{H}$ can be represented graphically by $|V(\mathcal{E}_3)|$ circles

C^x with $x \in V(\mathcal{E}_3)$ containing the variables $V(C^x) = \{x_1, \ldots, x_{n^x}\}$ as vertices. The edges are identified by the equations included in $\mathcal{H}$. The equations with exactly three variables are represented by hyperedges e with cardinality $|e| = 3$. The equations $x_i \oplus x_{i+1} = 0$ induce a circle containing the vertices $\{x_1, \ldots, x_{n^l}\}$ and the matching equations $x_i \oplus x_j = 0$ with $\{i, j\} \in M^x$ induce a perfect matching on the set of checker variables. An example of an instance of the Hybrid problem is depicted in Figure1.

In summary, we notice that there are four type of equations in the Hybrid problem (i) the circle equations $x_i \oplus x_{i+1} = 0$ with $i \in [7t_x - 1]$, (ii) circle border equations $x_1 \oplus x_{7t_x}$, (iii) matching equations $x_i \oplus x_j = 0$ with $\{i, j\} \in M^x$, and (iv) equations with three variables of the form $x \oplus y \oplus z = \{0, 1\}$.
In the remainder, we assume that equations of the form $x \oplus y \oplus z = \{0, 1\}$ contain only unnegated variables due to the transformation $\bar{x} \oplus y \oplus z = 0 \equiv x \oplus y \oplus z = 1$.

4 Our Contribution

We now formulate our main result.

Theorem 6. *Given an instance $\mathcal{H}$ of the Hybrid problem with n circles, m_2 equations with two variables and m_3 equations with exactly three variables with the properties described in Theorem 5, we construct in polynomial time an instance $\mathcal{S}_{\mathcal{H}}$ of the Shortest Superstring problem and Maximal Compression problem with the following properties:*

(i) *If there exists an assignment ϕ to the variables of $\mathcal{H}$ which leaves at most u equations unsatisfied, then, there exist a superstring s_ϕ for $\mathcal{S}_{\mathcal{H}}$ with length at most $|s_\phi| = 5m_2 + 22m_3 + 7n + u$.*

(ii) *From every superstring s for $S_{\mathcal{H}}$ with length $|s| = 5m_2 + 22m_3 + u + 7n$, we can construct in polynomial time an assignment ψ_s to the variables of $\mathcal{H}$ that leaves at most u equations in $\mathcal{H}$ unsatisfied.*

(iii) *If there exists an assignment ϕ to the variables of $\mathcal{H}$ which leaves at most u equations unsatisfied, then, there exist a superstring s_ϕ for $\mathcal{S}_{\mathcal{H}}$ with compression at least $comp(S_{\mathcal{H}}, s_\phi) = 3m_2 + 14m_3 - u + 5n$.*

(iv) *From every superstring s for $S_{\mathcal{H}}$ with compression $comp(S_{\mathcal{H}}, s) = 3m_2 + 14m_3 - u + 5n$, we can construct in polynomial time an assignment ψ_s to the variables of $\mathcal{H}$ that leaves at most u equations in $\mathcal{H}$ unsatisfied.*

(v) *The maximal orbit size of the instance $\mathcal{S}_{\mathcal{H}}$ is eight and the length of a string in $\mathcal{S}_{\mathcal{H}}$ is bounded by six.*

The former theorem can be used to derive an explicit approximation lower bound for the Shortest Superstring problem by reducing instances of the Hybrid problem of the form described in Theorem 5 to the Shortest Superstring problem.

Corollary 2. *For every $\epsilon > 0$, it is* $\mathbf{NP}$*-hard to approximate the Shortest Superstring problem with an approximation factor* $\frac{345}{344}$ $(1.0029) - \epsilon$.

Proof. First of all, we choose $k \in \mathbb{N}$ and $\delta > 0$ such that $\frac{345-\delta}{344+\delta+\frac{42}{k}} \geq \frac{345}{344} - \epsilon$ holds. Given an instance $\mathcal{E}_3$ of the MAX-E3-LIN problem, we generate k copies of $\mathcal{E}_3$ and produce an instance $\mathcal{H}$ of the Hybrid problem. Then, we construct the corresponding instance $\mathcal{S}_\mathcal{H}$ of the Shortest Superstring problem with the properties described in Theorem 6. We conclude according to Theorem 5 that there exist a superstring for $\mathcal{S}_\mathcal{H}$ with length at most

$$5 \cdot 60\nu k + 22 \cdot 2\nu k + \delta\nu k + 7n \leq (344 + \delta + \frac{7n}{k\nu})\nu k \leq (344 + \delta + \frac{7 \cdot 6}{k})\nu k$$

or the length of a superstring for $\mathcal{S}_\mathcal{H}$ is bounded from below by

$$5 \cdot 60\nu k + 22 \cdot 2\nu k + (1-\delta)\nu k + 7n \geq (344 + (1-\delta))\nu k \geq (345 - \delta)\nu k.$$

From Theorem 5, we know that the two cases above are $\mathbf{NP}$-hard to distinguish. Hence, for every $\epsilon > 0$, it is $\mathbf{NP}$-hard to find a solution to the Shortest Superstring problem with an approximation ratio $\frac{345-\delta}{344+\delta+\frac{42}{k}} \geq \frac{345}{344} - \epsilon$. $\square$

Analogously, Theorem 6 can be used to derive an approximation lower bound for the Maximal Compression problem.

Corollary 3. *For every $\epsilon > 0$, it is* $\mathbf{NP}$*-hard to approximate the Maximal Compression problem with an approximation factor* $\frac{207}{206}$ $(1.0048) - \epsilon$.

By applying Fact 1, we obtain the following hardness result for the MAX-ATSP problem.

Corollary 4. *For every $\epsilon > 0$, it is* $\mathbf{NP}$*-hard to approximate the MAX-ATSP problem with an approximation factor* $\frac{207}{206}$ $(1.0048) - \epsilon$.

5 Reduction from the Hybrid Problem

5.1 Main Ideas and Overview

Given an instance of the Hybrid problem $\mathcal{H}$, we want to transform $\mathcal{H}$ into an instance of the Shortest Superstring problem. Fortunately, the special structure of the linear equations in the Hybrid problem is particularly well-suited for our reduction, since a part of the equations with two variables form a circle and every variable occurs exactly three times. For every equation $g_{i+1} \equiv x_i \oplus x_{i+1} = 0$ included in this circle, we introduce a set $S(g_{i+1})$ containing two strings, which can be aligned advantageously in two natural ways. If those fragments corresponding to two successively following equations $x_{i-1} \oplus x_i = 0$ and $x_i \oplus x_{i+1} = 0$ use the same natural alignment, we are able to overlap those fragments by one letter. From a high level view, we can construct an associated superstring for each circle in $\mathcal{H}$, which contains the natural aligned strings. In fact, we define for every equation $g \in \mathcal{H}$ an associated set of strings $S(g)$ and the corresponding natural alignment. The instance $\mathcal{S}_{\mathcal{H}}$ of the Shortest Superstring problem is given by the union of all sets $S(g)$. Due to the construction of the sets $S(g)$, there is a particular way to interpret an alignment of the strings in $S(g)$ included in the resulting superstring as an assignment to the variables in the Hybrid instance. The major challenge in the proof of correctness is to prove that every superstring for $\mathcal{S}_{\mathcal{H}}$ can be interpreted as an assignment to the variables in the Hybrid instance $\mathcal{H}$ with the property that the number of satisfied equations is connected to the length of the superstring.

5.2 Constructing $\mathcal{S}_{\mathcal{H}}$ from $\mathcal{H}$

Given an instance of the Hybrid problem $\mathcal{H}$, we are going to construct the corresponding instance $\mathcal{S}_{\mathcal{H}}$ of the Shortest Superstring problem. Furthermore, we introduce some notations and conventions.

For every equation $g \in \mathcal{H}$, we define a set $S(g)$ of corresponding strings. The corresponding instance $\mathcal{S}_{\mathcal{H}}$ of the Shortest Superstring problem is given by $\mathcal{S}_{\mathcal{H}} = \bigcup\limits_{g \in \mathcal{H}} S(g)$. The strings in the set $S(g)$ differ by the type of considered equation $g \in \mathcal{H}$. Let us start with the description of $\mathcal{S}_{\mathcal{H}}$. Therefore, we need to specify the instance of the Hybrid problem more precisely.
Let $\mathcal{E}_3$ be an instance of the MAX-E3-LIN problem and $\mathcal{H}$ its corresponding instance of the Hybrid problem with n circles. For every variable $x \in V(\mathcal{E}_3)$, there is an associated circle C^x. Each circle consists of $m_2^x - 1$ circle equations g_{i+1}^x with $i \in [m_2^x - 1]$, a circle border equation $g_1^x \equiv x_1 \oplus x_{m_2^x} = 0$ and $|M^x|$ matching equations g_e^x with $e \in M^x$. Furthermore, we have m_3 equations g_j^3 with exactly three variables. We are going to specify the sets $S(g)$ differing by the type of equation g, whereby we distinguish four types of equations contained in $\mathcal{H}$.

- (i) circle equations
- (ii) matching equations
- (iii) circle border equations
- (iv) equations with exactly three variables

We begin with the description of the strings corresponding to circle border equations.

Strings Corresponding to Circle Border Equations

Given an instance of the hybrid problem $\mathcal{H}$, a circle C^x in $\mathcal{H}$ and its circle border equation $g_1^x \equiv x_1 \oplus x_n = 0$, we introduce six associated strings, that are all included in the set $S(g_1^x)$. Due to the construction of the circle C^x, the variable x_n is a contact variable. This means that x_n appears in an equation g_j^3 with exactly three variables. The strings in the set $S(g_1^x)$ differ by the type of equation g_j^3. We begin with the case $g_j^3 \equiv x_n \oplus y \oplus z = 0$.

The string $L_x C_x^l$ is used as the initial part of the superstring corresponding to this circle, whereas $C_x^r R_x$ is used as the end part. Furthermore, we introduce strings that represent an assignment that sets either the variable x_1 to 0 or the variable x_n to 1. The corresponding two strings are

$$C_x^l x_1^{m0} x_n^{l1} C_x^r \qquad \text{and} \qquad x_n^{l1} C_x^r C_x^l x_1^{m0}.$$

Finally, we define the last two strings of the set $S(g_1^x)$

$$C_x^l x_1^{r1} x_n^{m0} C_x^r \qquad \text{and} \qquad x_n^{m0} C_x^r C_x^l x_1^{r1}.$$

having a similar interpretation. Both pairs of strings can be overlapped by two letters. Those natural alignments have a crucial influence during the process of constructing a superstring. For this reason, we introduce a notation for this alignments. By the 0-*alignment* of the strings in $S(g_1^x)$, we refer to the following alignment of the four strings. In the following, $(\downarrow)$ will denote the overlapping of the strings.

$$\begin{array}{ccc} C_x^l x_1^{m0} x_n^{l1} C_x^r \quad \text{and} \quad x_n^{l1} C_x^r C_x^l x_1^{m0} & & C_x^l x_1^{r1} x_n^{m0} C_x^r \quad \text{and} \quad x_n^{m0} C_x^r C_x^l x_1^{r1}. \\ \downarrow & & \downarrow \\ C_x^l x_1^{m0} x_n^{m1} C_x^r C_x^l x_1^{m0} & \text{and} & x_n^{l0} C_x^r C_x^l x_1^{r1} x_n^{l0} C_x^r \end{array}$$

On the other hand, we the define the 1-*alignment* of the strings in $S(g_1^x)$ as follows.

$$\begin{array}{ccc} C_x^l x_1^{m0} x_n^{l1} C_x^r \quad \text{and} \quad x_n^{l1} C_x^r C_x^l x_1^{m0} & & C_x^l x_1^{r1} x_n^{m0} C_x^r \quad \text{and} \quad x_n^{m0} C_x^r C_x^l x_1^{r1}. \\ \downarrow & & \downarrow \\ x_n^{l1} C_x^r C_x^l x_1^{m0} x_n^{l1} C_x^r & \text{and} & C_x^l x_1^{r1} x_n^{m0} C_x^r C_x^l x_1^{r1} \end{array}$$

Both ways to join the four strings are called *simple* alignments.

After having described how the strings corresponding to $S(g_1^x)$ in case of $g_j^3 \equiv x_n \oplus y \oplus z = 0$ are defined, we are going to deal with the case $g_j^3 \equiv x_n \oplus y \oplus z = 1$.
As before, we use $L_x C_x^l$ as the initial part of the superstring corresponding to this circle, whereas $C_x^r R_x$ is used as the end part. Furthermore, we define the

remaining four strings contained in $S(g_1^x)$ by the following.

$$C_x^l x_1^{m0} x_n^{m1} C_x^r \qquad x_n^{m1} C_x^r C_x^l x_1^{m0}$$

and

$$C_x^l x_1^{r1} x_n^{l0} C_x^r \qquad x_n^{l0} C_x^r C_x^l x_1^{r1}$$

Both pairs of strings can be overlapped by two letters. We introduce a notation for this alignments.

$$\begin{array}{ccc} C_x^l x_1^{m0} x_n^{m1} C_x^r \quad \text{and} \quad x_n^{m1} C_x^r C_x^l x_1^{m0} & & C_x^l x_1^{r1} x_n^{l0} C_x^r \quad \text{and} \quad x_n^{l0} C_x^r C_x^l x_1^{r1}. \\ \downarrow & & \downarrow \\ C_x^l x_1^{m0} x_n^{m1} C_x^r C_x^l x_1^{m0} & \text{and} & x_n^{l0} C_x^r C_x^l x_1^{r1} x_n^{l0} C_x^r \end{array}$$

The former introduced alignment is called the 0-*alignment* of the strings in $S(g_1^x)$. On the other hand, we the define the 1-*alignment* of the strings in $S(g_1^x)$ as follows.

$$\begin{array}{ccc} C_x^l x_1^{m0} x_n^{m1} C_x^r \quad \text{and} \quad x_n^{m1} C_x^r C_x^l x_1^{m0} & & C_x^l x_1^{r1} x_n^{l0} C_x^r \quad \text{and} \quad x_n^{l0} C_x^r C_x^l x_1^{r1}. \\ \downarrow & & \downarrow \\ x_n^{m1} C_x^r C_x^l x_1^{m0} x_n^{m1} C_x^r & \text{and} & C_x^l x_1^{r1} x_n^{l0} C_x^r C_x^l x_1^{r1} \end{array}$$

In the remainder, we refer to both ways to overlap the four strings as *simple* alignments. Next, we describe the strings corresponding to matching equations.

Strings Corresponding to Matching Equations

Let C^x be a circle in $\mathcal{H}$ and M^x its associated perfect matching. Let $\{i, j\}$ be an edge in M^x and $g_{\{i,j\}}^x \equiv x_i \oplus x_j = 0$ the associated matching equation. We now define the corresponding set $S(g_{\{i,j\}}^x)$ consisting of two strings, whereby we assume $i < j$. Then, we introduce two strings of the form

$$x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1} \qquad \text{and} \qquad x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}$$

corresponding to the matching equation. There are two ways to align those two strings to obtain an overlap of two letters. In the remainder, we refer to those alignments as *simple*.

$$x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1} \quad \text{and} \quad x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}$$

$$\swarrow \qquad \searrow$$

$$\overbrace{x_j^{r0}x_j^{l0}\rlap{$\underbrace{\phantom{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}}_{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}$}x_i^{r1}x_i^{l1}}^{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}x_j^{r0}x_j^{l0} \qquad \overbrace{x_i^{r1}x_i^{l1}\rlap{$\underbrace{\phantom{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}}_{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}$}x_j^{r0}x_j^{l0}}^{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}x_i^{r1}x_i^{l1}$$

The first way to overlap the strings is called the 0-*alignment*, whereas the second one is called the 1-*alignment*. Next, we describe the strings corresponding to circle equations.

Strings Corresponding to Circle Equations

Let C^x be a circle in $\mathcal{H}$ and M^x its associated matching. Furthermore, let $\{i, j\}$ and $\{i+1, j'\}$ be both contained in M^x. We assume that $i < j$. Then, we introduce the corresponding strings for $x_i \oplus x_{i+1} = 0$. If $i+1 < j'$, we have

$$x_i^{m1}x_{i+1}^{m1}x_i^{r0}x_{i+1}^{l0} \qquad \text{and} \qquad x_i^{r0}x_{i+1}^{l0}x_i^{m1}x_{i+1}^{m1}.$$

Otherwise $(i+1 > j')$, we have

$$x_i^{m1}x_{i+1}^{l1}x_i^{r0}x_{i+1}^{m0} \qquad \text{and} \qquad x_i^{r0}x_{i+1}^{m0}x_i^{m1}x_{i+1}^{l1}.$$

In case of $i > j$ and $i+1 > j'$, we use

$$x_i^{r1}x_{i+1}^{l1}x_i^{m0}x_{i+1}^{m0} \qquad \text{and} \qquad x_i^{m0}x_{i+1}^{m0}x_i^{r1}x_{i+1}^{l1}$$

Finally, if $i > j$ and $i+1 < j'$, we introduce

$$x_i^{r1}x_{i+1}^{m1}x_i^{m0}x_{i+1}^{l0} \qquad \text{and} \qquad x_i^{m0}x_{i+1}^{l0}x_i^{r1}x_{i+1}^{m1}$$

Let x_i be a variable in $\mathcal{H}$ contained in an equation g_j^3 with three variables. We now define the corresponding strings for the equations $x_{i-1} \oplus x_i = 0$ and $x_i \oplus x_{i+1} = 0$. We assume that $\{i-1, j\}$ and $\{i+1, j'\}$ are both included in M^x. Furthermore, we assume $i-1 < j$ and $i+1 < j'$. If the equation g_j^3 is of the form $x_i \oplus y \oplus z = 0$, we introduce

$$x_{i-1}^{m1}x_i^{r1}x_{i-1}^{r0}x_i^{m0} \qquad \text{and} \qquad x_{i-1}^{r0}x_i^{m0}x_{i-1}^{m1}x_i^{r1}.$$

for $x_{i-1} \oplus x_i = 0$. Furthermore, for $x_i \oplus x_{i+1} = 0$, we use the strings

$$x_i^{l1}x_{i+1}^{m1}x_i^{m0}x_{i+1}^{l0} \qquad \text{and} \qquad x_i^{m0}x_{i+1}^{l0}x_i^{l1}x_{i+1}^{m1}.$$

On the other hand, if the equation g_j^3 is of the form $x_i \oplus y \oplus z = 1$, we introduce

$$x_{i-1}^{m1}x_i^{m1}x_{i-1}^{r0}x_i^{r0} \qquad \text{and} \qquad x_{i-1}^{r0}x_i^{r0}x_{i-1}^{m1}x_i^{m1}.$$

corresponding to the equation $x_{i-1} \oplus x_i = 0$. For $x_i \oplus x_{i+1} = 0$, we use the strings

$$x_i^{m1} x_{i+1}^{m1} x_i^{l0} x_{i+1}^{l0} \qquad \text{and} \qquad x_i^{l0} x_{i+1}^{l0} x_i^{m1} x_{i+1}^{m1}.$$

Accordingly, we introduce the notation of simple alignments for the strings in $S(g_{i+1}^x)$. For the strings

$$x_i^{m1} x_{i+1}^{m1} x_i^{r0} x_{i+1}^{l0} \qquad \text{and} \qquad x_i^{r0} x_{i+1}^{l0} x_i^{m1} x_{i+1}^{m1},$$

we define the following alignments as simple.

$$\overbrace{x_i^{m1} x_{i+1}^{m1} x_i^{r0} x_{i+1}^{l0}}^{x_i^{m1} x_{i+1}^{m1} x_i^{r0} x_{i+1}^{l0}}\!\underbrace{\phantom{x_i^{r0} x_{i+1}^{l0}} x_i^{m1} x_{i+1}^{m1}}_{x_i^{r0} x_{i+1}^{l0} x_i^{m1} x_{i+1}^{m1}} \qquad \text{and} \qquad \overbrace{x_i^{r0} x_{i+1}^{l0} x_i^{m1} x_{i+1}^{m1}}^{x_i^{r0} x_{i+1}^{l0} x_i^{m1} x_{i+1}^{m1}}\!\underbrace{\phantom{x_i^{m1} x_{i+1}^{m1}} x_i^{r0} x_{i+1}^{l0}}_{x_i^{m1} x_{i+1}^{m1} x_i^{r0} x_{i+1}^{l0}}$$

The the former alignment is called the 1-alignment and the latter one is called the 0-alignment. Next, we describe the strings corresponding to equations with three variables.

Strings Corresponding to Equations with Three Variables

We now concentrate on equations with exactly three variables. Let g_j^3 be an equation with three variables in $\mathcal{H}$. For every equation g_j^3, we define two corresponding sets $S^A(g_j^3)$ and $S^B(g_j^3)$, which both contain exactly three strings. Finally, the set $S(g_j^3)$ is defined by the union $S^A(g_j^3) \cup S^B(g_j^3)$. We distinguish whether g_j^3 is of the form $x \oplus y \oplus z = 1$ or $x \oplus y \oplus z = 0$. The description starts with the former case.

An equation of the form $x \oplus y \oplus z = 0$ is represented by $S^A(g_j^3)$ containing the strings

$$x^{r1} A_j^1 x^{l1} y^{r1} A_j^2 y^{l1} \qquad y^{r1} A_j^2 y^{l1} x^{m0} A_j^3 C_j \qquad x^{m0} A_j^3 C_j x^{r1} A_j^1 x^{l1}$$

and by $S^B(g_j^3)$ containing the strings

$$x^{r1} B_j^1 x^{l1} z^{r1} B_j^2 z^{l1} \qquad z^{r1} B_j^2 z^{l1} C_j B_j^3 x^{m0} \qquad C_j B_j^3 x^{m0} x^{r1} B_j^1 x^{l1}$$

On the other hand, for equations of the form $g_j^3 \equiv x \oplus y \oplus z = 1$, we introduce $S^A(g_j^3)$ containing the following strings.

$$x^{r0} A_j^1 x^{l0} y^{r0} A_j^2 y^{l0} \qquad y^{r0} A_j^2 y^{l0} x^{r1} A_j^3 C_j \qquad x^{r1} A_j^3 C_j x^{r0} A_j^1 x^{l0}$$

Furthermore, we give the definition of $S^B(g_j^3)$, which includes the following strings.

$$x^{r0} B_j^1 x^{l0} z^{r0} B_j^2 z^{l0} \qquad z^{r0} B_j^2 z^{l0} C_j B_j^3 x^{l1} \qquad C_j B_j^3 x^{l1} x^{r0} B_j^1 x^{l0}$$

The strings in the set $S^A(g_j^3)$ can be aligned in a cyclic fashion in order to obtain different strings which we will use in our reduction. Every specific alignment possesses its own abbreviation given below.

$$x^{r1}A_j^1x^{l1}y^{r1}A_j^2y^{l1} \quad y^{r1}A_j^2y^{l1}x^{m0}A_j^3C_j \quad x^{m0}A_j^3C_jx^{r1}A_j^1x^{l1}$$

$$\downarrow$$

$$\overbrace{x^{r1}A_j^1x^{l1}\mathrlap{\underbrace{\phantom{y^{r1}A_j^2y^{l1}x^{m0}A_j^3C_j}}_{y^{r1}A_j^2y^{l1}x^{m0}A_j^3C_j}}y^{r1}A_j^2y^{l1}}^{x^{r1}A_j^1x^{l1}y^{r1}A_j^2y^{l1}}\overbrace{x^{m0}A_j^3C_jx^{r1}A_j^1x^{l1}}^{x^{m0}A_j^3C_jx^{r1}A_j^1x^{l1}} \equiv x^{r1}A_jx^{l1} \text{ called } x^1\text{- alignment}$$

$$\downarrow$$

$$\underbrace{y^{r1}A_j^2y^{l1}\mathrlap{\overbrace{\phantom{x^{m0}A_j^3C_jx^{r1}A_j^1x^{l1}}}^{x^{m0}A_j^3C_jx^{r1}A_j^1x^{l1}}}x^{m0}A_j^3C_j}_{y^{r1}A_j^2y^{l1}x^{m0}A_j^3C_j}\underbrace{x^{r1}A_j^1x^{l1}y^{r1}A_j^2y^{l1}}_{x^{r1}A_j^1x^{l1}y^{r1}A_j^2y^{l1}} \equiv y^{r1}A_jy^{l1} \text{ called } y^1\text{- alignment}$$

$$\downarrow$$

$$\underbrace{x^{m0}A_j^3C_j\mathrlap{\overbrace{\phantom{x^{r1}A_j^1x^{l1}y^{r1}A_j^2y^{l1}}}^{x^{r1}A_j^1x^{l1}y^{r1}A_j^2y^{l1}}}x^{r1}A_j^1x^{l1}}_{x^{m0}A_j^3C_jx^{r1}A_j^1x^{l1}}\underbrace{y^{r1}A_j^2y^{l1}x^{m0}A_j^3C_j}_{y^{r1}A_j^2y^{l1}x^{m0}A_j^3C_j} \equiv x^{m0}A_jC_j \text{ called left-}x^0\text{- alignment}$$

Analogously, the strings in $S^B(g_j^3)$ can also be aligned in a cyclic fashion. We are going to define the abbreviations for those alignments.

$$x^{r1}B_j^1x^{l1}z^{r1}B_j^2z^{l1} \quad z^{r1}B_j^2z^{l1}C_jB_j^3x^{m0} \quad C_jB_j^3x^{m0}x^{r1}B_j^1x^{l1}$$

$$\downarrow$$

$$\underbrace{C_jB_j^3x^{m0}\mathrlap{\overbrace{\phantom{x^{r1}B_j^1x^{l1}z^{r1}B_j^2z^{l1}}}^{x^{r1}B_j^1x^{l1}z^{r1}B_j^2z^{l1}}}x^{r1}B_j^1x^{l1}}_{C_jB_j^3x^{m0}x^{r1}B_j^1x^{l1}}\underbrace{z^{r1}B_j^2z^{l1}C_jB_j^3x^{m0}}_{z^{r1}B_j^2z^{l1}C_jB_j^3x^{m0}} \equiv C_jB_jx^{m0} \text{ called right-}x^0\text{- alignment}$$

$$\downarrow$$

$$\overbrace{x^{r1}B_j^1x^{l1}\mathrlap{\underbrace{\phantom{z^{r1}B_j^2z^{l1}C_jB_j^3x^{m0}}}_{z^{r1}B_j^2z^{l1}C_jB_j^3x^{m0}}}z^{r1}B_j^2z^{l1}}^{x^{r1}B_j^1x^{l1}z^{r1}B_j^2z^{l1}}\overbrace{C_jB_j^3x^{m0}x^{r1}B_j^1x^{l1}}^{C_jB_j^3x^{m0}x^{r1}B_j^1x^{l1}} \equiv x^{r1}B_jx^{l1} \text{ called } x^1\text{- alignment}$$

$$\downarrow$$

$$\underbrace{z^{r1}B_j^2z^{l1}\mathrlap{\overbrace{\phantom{C_jB_j^3x^{m0}x^{r1}B_j^1x^{l1}}}^{C_jB_j^3x^{m0}x^{r1}B_j^1x^{l1}}}C_jB_j^3x^{m0}}_{z^{r1}B_j^2z^{l1}C_jB_j^3x^{m0}}\underbrace{x^{r1}B_j^1x^{l1}z^{r1}B_j^2z^{l1}}_{x^{r1}B_j^1x^{l1}z^{r1}B_j^2z^{l1}} \equiv z^{r1}B_jz^{l1} \text{ called } z^1\text{- alignment}$$

The strings in $S^B(g_j^3)$ and $S^A(g_j^3)$ can be overlapped in a special way that corresponds to assigning the value 0 to x.

$$x^{r1}A_j^1x^{l1}y^{r1}A_j^2y^{l1} \quad y^{r1}A_j^2y^{l1}x^{m0}A_j^3C_j \quad x^{m0}A_j^3C_jx^{r1}A_j^1x^{l1}$$
$$z^{r1}B_j^2z^{l1}C_jB_j^3x^{m0} \quad C_jB_j^3x^{m0}x^{r1}B_j^1x^{l1} \quad x^{r1}B_j^1x^{l1}z^{r1}B_j^2z^{l1}$$
$$\downarrow$$
$$x^{r1}A_j^1x^{l1}y^{r1}A_j^2y^{l1} \qquad C_jB_j^3x^{m0}x^{r1}B_j^1x^{l1} \qquad z^{r1}B_j^2z^{l1}C_jB_j^3x^{m0}$$
$$x^{r0}A_j^3C_jx^{r1}A_j^1x^{l1}y^{r1}A_j^2y^{l1}x^{m0}A_j^3C_jB_j^3x^{m0}x^{r1}B_j^1x^{l1}z^{r1}B_j^2z^{l1}C_jB_j^3x^{m0}$$
$$x^{r0}A_j^3C_jx^{r1}A_j^1x^{l1} \qquad y^{r1}A_j^2y^{l1}x^{m0}A_j^3C_j \qquad x^{r1}B_j^1x^{l1}z^{r1}B_j^2z^{l1}$$

In the remainder, we call this alignment the x^0-alignment of $S(g_j^3)$ and use the abbreviation $x^{m0}C_jx^{m0}$ for this string.

5.3 Constructing the Superstring s_ϕ from ϕ

Given an assignment ϕ to the variables of $\mathcal{H}$, we are going to construct the associated superstring s_ϕ for the instance $\mathcal{S}_\mathcal{H}$.

For every $g \in \mathcal{H}$, we formulate rules for aligning the corresponding strings in $S(g)$ according to the assignment ϕ. We start with sets corresponding to circle border equations and circle equations. Afterwards we show how the actual fragments can be overlapped with strings from the sets corresponding to matching equations and equations with three variables. Furthermore, we analyze the relation between the assignment ϕ and the length of the obtained superstring s_ϕ. We begin with the description of the alignment of strings corresponding to circle border equations in $\mathcal{H}$.

Aligning Strings Corresponding to Circle Border Equations

Let C^x be a circle in $\mathcal{H}$ and $x_1 \oplus x_n = 0$ its circle border equation. Furthermore, we assume that x_n is contained in a equation with three variables of the form $x_n \oplus y \oplus z = 0$. First, we set the string $L_xC_x^l$ as the initial part of our superstring corresponding to the circle C^x. Then, we use the $\phi(x_1)$-alignment of the strings

$$C_x^lx_1^{m0}x_n^{m1}C_x^r, \quad x_n^{m1}C_x^rC_x^lx_1^{m0}, \quad C_x^lx_1^{r1}x_n^{l0}C_x^r, \quad \text{and} \quad x_n^{l0}C_x^rC_x^lx_1^{r1}.$$

In this condition, one of the strings s_l can be overlapped from the left side with $L_xC_x^l$ by one letter. The other string s_r will be joined from the right side with $C_x^rR_x$ by one letter. This construction will help us to check whether the variable x_n is assigned the same value as x_1 by ϕ. The string s_r can be interpreted as the $\phi(x_1)$-alignment of the strings corresponding to $x_n \oplus x_{n+1} = 0$, since the first letter of s_r is either x_n^{m1} or x_n^{l0}.

The parts corresponding to a circle border equation with $x_n \oplus y \oplus z = 1$ can be constructed analogously. Next, we are going to align strings corresponding to circle equations.

Aligning Strings Corresponding to Circle Equations

Let $x_i \oplus x_{i+1} = 0$ be a circle equation contained in $\mathcal{H}$. Furthermore, let the corresponding strings are given by

$$x_i^{m0} x_{i+1}^{r0} x_i^{l1} x_{i+1}^{m1} \quad \text{and} \quad x_i^{l1} x_{i+1}^{m1} x_i^{m0} x_{i+1}^{r0}.$$

In dependence of the given assignment ϕ, we use simple alignments to overlap the considered strings. More precisely, we make use of the $\phi(x_{i+1})$-alignment. For every pair of associated strings, we derive an overlap of two letters. We are going to align those fragments with strings corresponding to matching equations and equations with three variables.

Aligning Strings Corresponding to Matching Equations

Let $x_i \oplus x_j = 0$ be a matching equation in $\mathcal{H}$. Let us assume that $i < j$. We define the alignment of the strings in $S(g^x_{\{i,j\}})$ according to the value of $\phi(x_{i+1})$. More precisely, we use the $\phi(x_{i+1})$-alignment of the strings

$$x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1} \quad \text{and} \quad x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}.$$

Due to this alignment, we obtain an overlap of two letters. We are going to analyze the length of the resulting superstring in dependence of the assignment ϕ to the variables x_i, x_{i+1}, x_j and x_{j+1}. We start with the case $\phi(x_{i+1}) = \phi(x_{j+1}) = 1$.

Case $\phi(x_{i+1}) = \phi(x_{j+1}) = 1$:
We use the 1-alignment of the strings $x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}$ and $x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1}$. The situation is depicted below. (The two triangle notation $\rhd\rhd$ and $\lhd\lhd$ will be explained hereafter.)

$$b \rhd\rhd \boxed{X_i}\, x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1}\, \boxed{x_i^{l1}} \lhd\lhd m \rhd\rhd \boxed{Y_j}\, x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}\, \boxed{x_j^{m1}} \lhd\lhd e$$

$$\downarrow$$

$$b \rhd\rhd \boxed{X_i}\, \overbrace{x_i^{r1} x_i^{l1} \rlap{$\underbrace{\phantom{x_j^{r0} x_j^{l0} x_i^{r1} \boxed{x_i^{l1}}}}_{x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1}}$} x_j^{r0} x_j^{l0}}^{x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}} x_i^{r1} \boxed{x_i^{l1}} \lhd\lhd m \rhd\rhd \boxed{Y_j}\, \boxed{x_j^{m1}} \lhd\lhd e$$

The actual superstring s is denoted by the following sequence.

$$s = b \rhd\rhd \boxed{X_i}\, x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1} \boxed{x_i^{l1}} \lhd\lhd\ m \rhd\rhd \boxed{Y_j}\, x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0} \boxed{x_j^{m1}} \lhd\lhd\ e$$

The part $\rhd\rhd \boxed{X_i}$ represents a simple alignment of the strings corresponding to $x_{i-1} \oplus x_i = 0$ ending with the letter $X_i \in \{x_i^{m0}, x_i^{r1}\}$, which means

$$\rhd\rhd \boxed{X_i} \in \{x_{i-1}^{m0}x_i^{m0}x_{i-1}^{l1}x_i^{r1}x_{i-1}^{m0}x_i^{m0}, x_{i-1}^{l1}x_i^{r1}x_{i-1}^{m0}x_i^{m0}x_{i-1}^{l1}x_i^{r1}\}.$$

The letter in the box emphasizes the letter which can be used to overlap from the right side with other strings. Furthermore, the string $\boxed{x_i^{l1}} \lhd\lhd$ denotes $x_i^{l1}x_{i+1}^{r1}x_i^{m0}x_{i+1}^{m0}x_i^{l1}x_{i+1}^{r1}$. Analogously, $\rhd\rhd \boxed{Y_j}$ is a simple alignment of the strings corresponding to $x_{j-1} \oplus x_j = 0$, whereby $Y_j \in \{x_j^{r0}, x_j^{m1}\}$. Furthermore, we use $\boxed{x_j^{m1}} \lhd\lhd$ to denote $x_j^{m1}x_{j+1}^{m1}x_j^{l0}x_{j+1}^{r0}x_j^{m1}x_{j+1}^{m1}$. Finally, b, m and e are sequences of letters, which we do not specify in detail. They define the remaining parts of the superstring s.
If $X_i = x_i^{r1}$ holds, we align $\rhd\rhd \boxed{X_i}$ with $x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}$ to achieve an additional overlap of one letter. An analogue situation holds for $\rhd\rhd \boxed{Y_j}$ and $\boxed{x_j^{m1}} \lhd\lhd$. All in all, we obtain an overlap of three letters if $\phi(x_i) = \phi(x_{i+1}) = 1$ and $\phi(x_{j+1}) = \phi(x_j) = 1$ holds. Otherwise, we lose an overlap of one letter per unsatisfied equation.

Case $\phi(x_{i+1}) = \phi(x_{j+1}) = 0$:
We use the 0-alignment of the strings $x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}$ and $x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}$.

$$b \rhd\rhd \boxed{X_i}\, x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1} \boxed{x_i^{m0}} \lhd\lhd\ m \rhd\rhd \boxed{Y_j}\, x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0} \boxed{x_j^{l0}} \lhd\lhd\ e$$

$$\downarrow$$

$$b \rhd\rhd \boxed{X_i}\boxed{x_i^{m0}} \lhd\lhd\ m \rhd\rhd \boxed{Y_j}\, \rlap{$\overbrace{\phantom{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}}^{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}$}x_j^{r0}x_j^{l0}\underbrace{x_i^{r1}x_i^{l1}x_j^{r0}\boxed{x_j^{l0}}}_{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}} \lhd\lhd\ e$$

In this case, we use $\boxed{x_i^{m0}} \lhd\lhd$ as an abbreviation for $x_i^{m0}x_{i+1}^{m0}x_i^{l1}x_{i+1}^{r1}x_i^{m0}x_{i+1}^{m0}$ and $\boxed{x_j^{l0}} \lhd\lhd$ for $x_j^{l0}x_{j+1}^{r0}x_j^{m1}x_{j+1}^{m1}x_j^{l0}x_{j+1}^{r0}$. If $X_i = x_i^{m0}$ holds, we align $\rhd\rhd \boxed{X_i}$ with $\boxed{x_i^{m0}} \lhd\lhd$ and gain an additional overlap of one letter. An analogue situation holds for $\rhd\rhd \boxed{Y_j}$ and $\boxed{x_j^{l0}} \lhd\lhd$. Hence, we obtain an overlap of three letters if $\phi(x_{i+1}) = \phi(x_i) = 0$ and $\phi(x_{j+1}) = \phi(x_j) = 0$ holds. If the corresponding equation with two variables is not satisfied, we lose an overlap of one letter.

Case $\phi(x_{i+1}) \neq \phi(x_{j+1}) = 1$**:**
In this case, we use the 0-alignment of the strings $x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}$ and $x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}$.

$$b \rhd\rhd \boxed{X_i}\, x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1} \boxed{x_i^{m0}} \lhd\lhd\ m \rhd\rhd \boxed{Y_j}\, x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0} \boxed{x_j^{m1}} \lhd\lhd\ e$$

$$\downarrow$$

$$b \rhd\rhd \boxed{X_i}\boxed{x_i^{m0}} \lhd\lhd\ m \rhd\rhd \boxed{Y_j}\boxed{x_j^{m1}} \lhd\lhd\ e \quad \rlap{$\overbrace{\phantom{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}}^{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}$}x_j^{r0}x_j^{l0}\underbrace{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}_{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}$$

We attach $x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}$ at the end of our actual solution s_ϕ without having any overlap with the so far obtained superstring. Notice that we obtain in each case an additional overlap of one letter if the corresponding equation with two variables is satisfied, i.e. $X_i = x_i^{m0}$ and $Y_j = x_j^{m1}$.

Case $\phi(x_{i+1}) \neq \phi(x_{j+1}) = 0$**:**
According to ϕ, we use the 1-alignment of the strings $x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}$ and $x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}$.

$$b \rhd\rhd \boxed{X_i}\, x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1} \boxed{x_i^{l1}} \lhd\lhd\ m \rhd\rhd \boxed{Y_j}\, x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0} \boxed{x_j^{l0}} \lhd\lhd\ e$$

$$\downarrow$$

$$b \rhd\rhd \boxed{X_i}\rlap{$\overbrace{\phantom{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}}^{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}$}x_i^{r1}x_i^{l1}\underbrace{x_j^{r0}x_j^{l0}x_i^{r1}\boxed{x_i^{l1}}}_{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}} \lhd\lhd\ m \rhd\rhd \boxed{Y_j}\boxed{x_j^{l0}} \lhd\lhd\ e$$

We join $x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}$ from the right side with $\boxed{x_i^{l1}}$ $\lhd\lhd$ and obtain an overlap of one letter. This reduces the length of the superstring by one letter independent of the assignment $\phi(x_j)$. In case of $X_i = x_i^{r1}$, we achieve another overlap of one letter, since we are able to align $\rhd\rhd$ $\boxed{X_i}$ from the right side with $x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}$. It corresponds to the satisfied equation $x_i \oplus x_{i+1} = 0$. Hence, we obtain at least the same number of overlapped letters as satisfied equations.

In summary, we note that we are able to achieve an overlap of at least one letter in each case if the corresponding equation is satisfied by ϕ. Hence, we obtain an overlap of at most three letters.

The other cases concerning equations $x_i \oplus x_j = 0$ with $i > j$ can be analyzed analogously. Next, we are going to align strings corresponding to equations with three variables.

Aligning Strings Corresponding to Equations with Three Variables

Let $g_j^3 \in \mathcal{H}$ be an equation with three variables x, y and z. Furthermore, let $x_{i-1} \oplus x = 0$, $x \oplus x_{i+1} = 0$, $y_{j-1} \oplus y = 0$, $y \oplus y_{j+1} = 0$, $z_{k-1} \oplus z = 0$ and $z \oplus z_{k+1} = 0$ be the equations with two variables, in which the variables x, y and z occur. Given the assignment ϕ to x, y and z, we are going to define the alignment of the corresponding strings. Let us start with equations of the form $g_j^3 \equiv x \oplus y \oplus z = 0$. Then, we define the rule for aligning strings in $S^A(g_j^3)$ and $S^B(g_j^3)$ as follows, whereby we handle the cases $\phi(x_{i+1}) + \phi(y_{j+1}) + \phi(z_{k+1}) = \{3, 2, 1, 0\}$ separately starting with $\phi(x_{i+1}) + \phi(y_{j+1}) + \phi(z_{k+1}) = 3$.

Case $\phi(x_{i+1}) + \phi(y_{j+1}) + \phi(z_{k+1}) = 3$**:**
In this case, we align the strings in $S(g_j^3)$ in such a way that we obtain the former introduced strings $y^{r1}A_jy^{l1}$ and $z^{r1}B_jz^{l1}$. The situation, which we want to analyze is depicted below.

$$b \rhd\rhd \boxed{X}\boxed{x^{l1}} \lhd\lhd\ m_1 \rhd\rhd \boxed{Y}\ y^{r1}A_jy^{l1}\ \boxed{y^{l1}} \lhd\lhd\ m_2 \rhd\rhd \boxed{Z}\ z^{r1}B_jz^{l1}\ \boxed{z^{l1}} \lhd\lhd\ e$$

$$\downarrow$$

$$b \rhd\rhd \boxed{X}\boxed{x^{l1}} \lhd\lhd\ m_1 \rhd\rhd \boxed{Y}\ \underbrace{y^{r1}A_j\boxed{y^{l1}}}_{y^{r1}A_jy^{l1}} \lhd\lhd\ m_2 \rhd\rhd \boxed{Z}\ \overbrace{z^{r1}B_j\boxed{z^{l1}}}^{z^{r1}B_jz^{l1}} \lhd\lhd\ e$$

Similarly to the situations that we discussed concerning the matching equations, we define the actual superstring s in the way described below.

$$s = b \rhd\rhd \boxed{X}\boxed{x^{l1}} \lhd\lhd\ m_1 \rhd\rhd \boxed{Y}\, y^{r1}A_jy^{l1} \boxed{y^{l1}} \lhd\lhd\ m_2 \rhd\rhd \boxed{Z}\, z^{r1}B_jz^{l1} \boxed{z^{l1}} \lhd\lhd\ e$$

Here, b, m_1, m_2 and e denote parts of s, which we do not specify in detail to emphasize the parts corresponding to the equation with three variables.
The string $\boxed{x^{l1}} \lhd\lhd$ denotes the $\phi(x_{i+1})$-alignment of the corresponding strings in $S(g_{i+1}^x)$. The strings $\boxed{z^{l1}} \lhd\lhd$ and $\boxed{y^{l1}} \lhd\lhd$ are defined analogously. In this situation, we want to analyze the cases $X \in \{x^{r1}, x^{m0}\}$, $Y \in \{y^{r1}, y^{m0}\}$ and $Z \in \{z^{r1}, z^{m0}\}$. We infer that we obtain an overlap of four letters if all equations with two variables are satisfied. Otherwise, we lose an overlap of one letter per unsatisfied equation with two variables.

Case $\phi(x_{i+1}) + \phi(y_{j+1}) + \phi(z_{k+1}) = 2$**:**
Let $\alpha, \gamma \in \{x_{i+1}, y_{j+1}, z_{k+1}\}$ be variables such that $\phi(\gamma) = \phi(\alpha) = 1$ holds. Then, we

use the α^1-alignment and γ^1-alignment of the strings in $S^A(g_j^3)$ and $S^B(g_j^3)$ breaking ties arbitrary. We display exemplary the situation for $\phi(z_{k+1}) = \phi(x_{i+1}) = 1$.

$$b \rhd\rhd \boxed{X}\, x^{r1}A_jx^{l1}\boxed{x^{l1}} \lhd\lhd\ m_1 \rhd\rhd \boxed{Y}\boxed{y^{m0}} \lhd\lhd\ m_2 \rhd\rhd \boxed{Z}\, z^{r1}B_jz^{l1}\boxed{z^{l1}} \lhd\lhd\ e$$

$$\downarrow$$

$$b \rhd\rhd \boxed{X}\ \underbrace{x^{r1}A_j\boxed{x^{l1}}}_{x^{r1}A_jx^{l1}} \lhd\lhd\ m_1 \rhd\rhd \boxed{Y}\ \boxed{y^{m0}} \lhd\lhd\ m_2 \rhd\rhd \boxed{Z}\ \overbrace{z^{r1}B_j\boxed{z^{l1}}}^{z^{r1}B_jz^{l1}} \lhd\lhd\ e$$

In this case, we achieve an overlap of five letters if all equations with two variables are satisfied. Otherwise, we lose an overlap of one letter per unsatisfied equation with two variables.

Case $\phi(x_{i+1}) + \phi(y_{j+1}) + \phi(z_{k+1}) = 1$**:**
If $\phi(z_{k+1}) + \phi(x_{i+1}) = 1$ holds, we align the strings in $S^B(g_j^3)$ and $S^A(g_j^3)$ to obtain $x^{r1}A_jx^{l1}$ and $z^{r1}B_jz^{l1}$. Otherwise, we make use of the strings $x^{r1}B_jx^{l1}$ and $y^{r1}A_jy^{l1}$. We display the situation for $\phi(y_{j+1}) = 1$.

$$b \rhd\rhd \boxed{X}\, x^{r1}B_jx^{l1}\boxed{x^{m0}} \lhd\lhd\ m_1 \rhd\rhd \boxed{Y}\, y^{r1}A_jy^{l1}\boxed{y^{l1}} \lhd\lhd\ m_2 \rhd\rhd \boxed{Z}\boxed{z^{m0}} \lhd\lhd\ e$$

$$\downarrow$$

$$b \rhd\rhd \boxed{X}\boxed{x^{m0}} \lhd\lhd\ m_1 \rhd\rhd \boxed{Y}\overbrace{y^{r1}A_j\boxed{y^{l1}}}^{y^{r1}A_jy^{l1}} \lhd\lhd\ m_2 \rhd\rhd \boxed{Z}\boxed{z^{m0}} \lhd\lhd\ e\, x^{r1}B_jx^{l1}$$

Notice that we obtain an overlap of four letters if the equations with two variables are satisfied, i.e. $X = x^{m0}$, $Z = z^{m0}$ and $Y = y^{r1}$. Otherwise, we lose an overlap of one letter per unsatisfied equation with two variables.

Case $\phi(x_{i+1}) + \phi(y_{j+1}) + \phi(z_{k+1}) = 0$**:**
In this case, we use the x^0-alignment of the strings in $S(g_j^3)$. The situation is displayed below.

$$b \rhd\rhd \boxed{X}\, x^{m0} C_j x^{m0} \boxed{x^{m0}} \lhd\lhd m_1 \rhd\rhd \boxed{Y}\boxed{y^{m0}} \lhd\lhd m_2 \rhd\rhd \boxed{Z}\boxed{z^{m0}} \lhd\lhd e$$

$$\downarrow$$

$$b \rhd\rhd \boxed{X}\, \underbrace{x^{m0} C_j \boxed{x^{m0}}}_{x^{m0} C_j x^{m0}} \lhd\lhd m_1 \rhd\rhd \boxed{Y}\boxed{y^{m0}} \lhd\lhd m_2 \rhd\rhd \boxed{Z}\boxed{z^{m0}} \lhd\lhd e$$

Here, we are able to achieve an overlap of five letters if all equations with two variables are satisfied, i.e. $X = x^{m0}$, $Z = z^{m0}$ and $Y = y^{m0}$.

In summary, we state that we can achieve an overlap of at least one letter independent of the assignment ϕ. Additionally, we gain another overlap of one letter if the corresponding equation is satisfied by ϕ.

The situation for equations of the form $x \oplus y \oplus z = 1$ can be analyzed analogously. We are going to define the assignment ψ_s, which is associated to a given superstring for $\mathcal{S}_\mathcal{H}$.

5.4 Defining the Assignment

Given a superstring s for $\mathcal{S}_\mathcal{H}$, we are going to define the associated assignment ψ_s to the variables of $\mathcal{H}$. In order to deduce the values assigned to the variables in $\mathcal{H}$ from s, we have to normalize the given superstring s. For this reason, we define rules that transform a superstring for $\mathcal{S}_\mathcal{H}$ into a normed superstring for $\mathcal{S}_\mathcal{H}$ without increasing the length.

First, we introduce the definition of a normed superstring for $\mathcal{S}_\mathcal{H}$.

Definition 5 (Normed Superstring s for $\mathcal{S}_\mathcal{H}$)**.** *Let $\mathcal{H}$ be an instance of the Hybrid problem, $\mathcal{S}_\mathcal{H}$ the corresponding instance of the Shortest Superstring problem and s a superstring for $\mathcal{S}_\mathcal{H}$. We refer to s as a normed superstring for $\mathcal{S}_\mathcal{H}$ if for every $g \in \mathcal{H}$, the superstring s contains s_g as a proper substring, whereby s_g is resulted due to a simple alignment of the strings included in $S(g)$.*

After having defined a normed superstring, we are going to state rules which transform a superstring for $\mathcal{S}_\mathcal{H}$ into a normed superstring for $\mathcal{S}_\mathcal{H}$ without increasing the length of the underlying superstring. All transformation can be performed in polynomial time. Once accomplished to generate a normed superstring, we are able to define the assignment ψ_s and analyze the number of overlapped letters in dependence of the number of satisfied equations in $\mathcal{H}$ by ψ_s. Let us start with transformations of strings corresponding to circle equations and circle border equations.

Normalizing Strings Corresponding to Circle and Circle Border Equations

Let $x_i \oplus x_{i+1} = 0$ be a circle equation in $\mathcal{H}$. Furthermore, let $x_i^{m0}x_{i+1}^{r0}x_i^{l1}x_{i+1}^{m1}$ and $x_i^{l1}x_{i+1}^{m1}x_i^{m0}x_{i+1}^{r0}$ be its corresponding strings. We observe that those strings can have an overlap of at most one letter from the left side as well as from the right side with other strings in $\mathcal{S}_{\mathcal{H}}$. Given a superstring s for $\mathcal{S}_{\mathcal{H}}$, we obtain at least the same number of overlapped letters if we use one of the simple alignments in s. In particular, we have to use the simple alignment that maximizes the number of overlapped letters.

Given a superstring s for $\mathcal{S}_{\mathcal{H}}$, we separate the strings $x_i^{m0}x_{i+1}^{r0}x_i^{l1}x_{i+1}^{m1}$ and $x_i^{l1}x_{i+1}^{m1}x_i^{m0}x_{i+1}^{r0}$ from s. Consequently, this results in at most three strings bx_i^{m0}, $x_{i+1}^{m1}mx_i^{l1}$ and $x_{i+1}^{r0}e$ such that

$$s = bx_i^{m0}x_{i+1}^{r0}x_i^{l1}x_{i+1}^{m1}mx_i^{l1}x_{i+1}^{m1}x_i^{m0}x_{i+1}^{r0}e.$$

Then, we define the transformed superstring s' with at least the same number of overlapped letters by

$$s' = bx_i^{m0}x_{i+1}^{r0}x_i^{l1}x_{i+1}^{m1}x_i^{m0}x_{i+1}^{r0}e\,x_{i+1}^{m1}mx_i^{l1}.$$

In order to define the simple alignment, which is used in s by the strings in $S(g_{i+1}^x)$, we are going to state a criterion.

Let s be a superstring for $\mathcal{S}_{\mathcal{H}}$ and g_{i+1}^x a circle equation. Let the corresponding strings are given by $x_i^{m0}x_{i+1}^{r0}x_i^{l1}x_{i+1}^{m1}$ and $x_i^{l1}x_{i+1}^{m1}x_i^{m0}x_{i+1}^{r0}$. Then, we say that the strings in $S(g_{i+1}^x)$ use a 1-alignment in s if there are more strings s^1 in $\mathcal{S}_{\mathcal{H}}\backslash S(g_{i+1}^x)$ such that either s^1 is overlapped by one letter from the right side with $x_i^{l1}x_{i+1}^{m1}x_i^{m0}x_{i+1}^{r0}$ or s^1 is overlapped by one letter from the left side with $x_i^{m0}x_{i+1}^{r0}x_i^{l1}x_{i+1}^{m1}$ in s than strings s^0 in $\mathcal{S}_{\mathcal{H}}\backslash S(g_{i+1}^x)$ such that either s^0 is overlapped by one letter from the left side with $x_i^{l1}x_{i+1}^{m1}x_i^{m0}x_{i+1}^{r0}$ or s^0 is overlapped by one letter from the right side with $x_i^{m0}x_{i+1}^{r0}x_i^{l1}x_{i+1}^{m1}$ in s. Otherwise, the strings in $S(g_{i+1}^x)$ use a 0-alignment in s.

Given a superstring s for $\mathcal{S}_{\mathcal{H}}$, we define informally a part of the backbone of our transformed superstring by the strings s_g, whereby s_g is resulted due to a simple alignment used in s of the strings $S(g)$ for every circle equation $g \in \mathcal{H}$. Afterwards, we use this construction to align them with strings corresponding to matching equations, equations with three variables and circle border equations. Moreover, it will help us to define the assignment ψ_s and relate the number of satisfied equations to the number of overlapped letters. But first, we concentrate on circle border equations.

Let $x_1 \oplus x_n = 0$ be a circle border equation. Furthermore, let the corresponding strings are given by

$$L_xC_x^l, \quad C_x^lx_1^{m0}x_n^{l1}C_x^r, \quad x_n^{l1}C_x^rC_x^lx_1^{m0}, \quad C_x^lx_1^{r1}x_n^{m0}C_x^r, \quad x_n^{m0}C_x^rC_x^lx_1^{r1}, \quad \text{and} \quad C_x^rR_x.$$

Since the simple alignments of the strings in $S(g_1^x)$ achieve an overlap of two letters for each pair $\{C_x^l x_1^{m0} x_n^{l1} C_x^r,\ x_n^{l1} C_x^r C_x^l x_1^{m0}\}$ and $\{C_x^l x_1^{r1} x_n^{m0} C_x^r,\ x_n^{m0} C_x^r C_x^l x_1^{r1}\}$, we argue as before that those strings can be rearranged in a given superstring for $\mathcal{S}_{\mathcal{H}}$ such that the pairs use a simple alignment without increasing the length of the underlying superstring for $\mathcal{S}_{\mathcal{H}}$. In this situation, we are able to overlap one of the pairs using a simple alignment with $L_x C_x^l$ from the left side and the other one with $C_x^r R_x$ from the right side without increasing the length. This construction checks whether the variables x_1 and x_n have the same assigned value, which is rewarded by another overlap of one letter of the corresponding strings using a simple alignment.

For any fixed order of the circles C^x in $\mathcal{H}$, we build the backbone of our superstring consisting of the concatenation of the strings $s^x s^y \cdots s^z$, whereby a string s^x is associated to its circle C^x. Furthermore, s^x consists of the corresponding simple alignments of the strings in $S(g_i^x)$ used in s, whereby the order of the strings is given by the order of the variables in C^x. The string s^x starts with the letter L_x and ends with R_x.

Notice that similar transformations can be applied to strings corresponding to matching equations and to equations with three variables, but we are going to define the transformation for those strings in detail while analyzing the upper bound of overlapped letters for simple aligned strings corresponding to circle equations, which are contained in a given superstring s for $\mathcal{S}_{\mathcal{H}}$.

Before we start our analysis, we define the assignment ψ_s based on the actual superstring s for $\mathcal{S}_{\mathcal{H}}$, which is not necessarily a normed superstring for $\mathcal{S}_{\mathcal{H}}$. By applying the transformations, which we are going to define, the assignment ψ_s will change in dependence to the actual considered superstring.

$$
\begin{aligned}
\psi_s(x_i) &= 1 \text{ if the strings in } S(g_i^x) \text{ use a } 1\text{-alignment in } s \\
&= 0 \text{ otherwise}
\end{aligned}
$$

Due to the transformations for the strings corresponding to circle and circle border equations, the assignment ψ_s is well-defined.

Defining the Assignment for Checker Variables

Let $x \in V(\mathcal{E}_3)$, C^x be the corresponding circle in $\mathcal{H}$ and M^x its associated perfect matching. Furthermore, let $x_i \oplus x_{i+1} = 0$, $x_{i-1} \oplus x_i = 0$, $x_{j-1} \oplus x_j = 0$, $x_j \oplus x_{j+1} = 0$ and $x_i \oplus x_j = 0$ be equations in $\mathcal{H}$, whereby $\{i, j\} \in M^x$ and $i < j$ holds. Let s be a superstring for $\mathcal{S}_{\mathcal{H}}$ such that the strings corresponding to circle and circle border equations are using a simple alignment in s. Based on the the simple alignments of the strings corresponding to g_i^x, g_{i+1}^x, g_j^x and g_{j+1}^x, which are used in the superstring s, we are going to define the assignment to the variables x_i and x_j. Furthermore, we analyze the number of overlapped letters that can be achieved given the simple aligned strings and relate them to the number of satisfied equations in $\mathcal{H}$ by ψ_s.

In the remainder, we will assume that the underlying superstring for $\mathcal{S}_\mathcal{H}$ contains simple aligned strings corresponding to circle and circle border equations. Before we start our analysis, we introduce the notation of a constellation that denotes which of the simple alignments are used by the strings corresponding to the equations g_i^x, g_{i+1}^x, g_j^x and g_{j+1}^x in s.

Given a superstring s for $S_\mathcal{H}$ and $\{i,j\} \in M^x$, a *constellation* c is defined by $(X_i X_{i+1}, X_{j-1} X_{j+1})_{\{i,j\}}^s$ with $X_i, X_{i+1}, X_j, X_{j+1} \in \{0,1\}$, whereby $X_k = 1$ if and only if the strings in $S(g_k^x)$ use the 1-alignment in s for $k \in \{i, i+1, j, j+1\}$. We call a constellation c inconsistent if there is an entry $A_1 A_2$ with $A_1 \neq A_2$. Otherwise, c is called consistent.

Based on the given constellations, we are going to define ψ_s.

Definition 6 (Assignment ψ_s to Checker Variables)**.** *Let $\mathcal{H}$ be an instance of the Hybrid problem, $\mathcal{S}_\mathcal{H}$ its corresponding instance of the superstring problem and s a superstring for $\mathcal{S}_\mathcal{H}$. Given the constellation $(X_i X_{i+1}, X_j X_{j+1})_{\{i,j\}}^s$, we define ψ_s in the following way.*

- (i) *$\psi_s(x_i) = X_i$ and $\psi_s(x_j) = X_j$ if $X_i \oplus X_j = 1$ and c is consistent*
- (ii) *$\psi_s(x_i) = X_i$ and $\psi_s(x_j) = X_j$ if $X_i \oplus X_j = 0$*
- (iii) *$\psi_s(x_i) = 1 - X_i$ and $\psi_s(x_j) = X_j$ if $X_i \oplus X_j = 1$ and $X_i \neq X_{i+1}$*
- (iv) *$\psi_s(x_i) = X_i$ and $\psi_s(x_j) = 1 - X_j$ if $X_i \oplus X_j = 1$, $X_j \neq X_{j+1}$ and $X_i^1 = X_{i+1}$*

We are going to analyze the the different constellations and discuss the cases (i)-(iv) of the definition of ψ_s. We start with case (i).

CASE (i) $X_i \oplus X_j = 1$ and c is consistent:
There are two constellations, which we have to analyze, namely $(11,00)_{\{i,j\}}^s$ and $(00,11)_{\{i,j\}}^s$. Starting with the former constellation, we obtain the scenario depicted below. The string $\rhd\rhd\ \boxed{X_i}$ with $X_i \in \{x_i^{m0}, x_i^{r1}\}$ represents a simple alignment of the strings in $S(g_i^x)$. Analogously, the string

$$\boxed{X_{i+1}}\ \lhd\lhd \quad \text{with } X_{i+1} \in \{x_i^{m0}, x_i^{l1}\}$$

represents a simple alignment of the strings in $S(g_{i+1}^x)$. Since we know that using the most profitable simple alignment of the strings in $S(g_{\{i,j\}}^x)$ does not increase the length of the superstring, we make use of the 1-alignment and transform the superstring s in the superstring s', which are both depicted below.

$$s = b \triangleright\triangleright \boxed{x_i^{r1}}\, x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0} \boxed{x_i^{l1}} \triangleleft\triangleleft\ m \triangleright\triangleright \boxed{x_j^{r0}}\, x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1} \boxed{x_j^{l0}} \triangleleft\triangleleft\ e$$

$$\downarrow$$

$$s' = b \triangleright\triangleright \mathrlap{\overbrace{\phantom{\boxed{x_i^{r1}}\, x_i^{l1} x_j^{r0} x_j^{l0}}}^{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}}\boxed{x_i^{r1}}\, x_i^{l1} \underbrace{x_j^{r0} x_j^{l0} x_i^{r1} \boxed{x_i^{l1}}}_{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}} \triangleleft\triangleleft\ m \ \triangleright\triangleright \boxed{x_j^{r0}}\boxed{x_j^{l0}} \triangleleft\triangleleft\ e$$

Let us analyze the upper bound on the overlap of letters, whereby we are interested in the overlap of letters, which are additional to the overlap of two letters due to the simple alignment. In both cases, either by using the 1-alignment or the 0-alignment of the strings in $S(g^x_{\{i,j\}})$, we cannot obtain more than an overlap of two letters. It corresponds to the number of satisfied equations, which are $x_i \oplus x_{i+1} = 0$ and $x_j \oplus x_{j+1} = 0$.

In case of the constellation $(00, 11)^s_{\{i,j\}}$, we separate the strings $x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}$ and $x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}$ from the superstring s. Then, we attach the aligned string $x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}$ at the end of the actual solution. The considered situation is depicted below.

$$b \triangleright\triangleright \boxed{x_i^{m0}}\, x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0} \boxed{x_i^{m0}} \triangleleft\triangleleft\ m \triangleright\triangleright \boxed{x_j^{m1}}\, x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1} \boxed{x_j^{m1}} \triangleleft\triangleleft\ e$$

$$\downarrow$$

$$\overbrace{b \triangleright\triangleright \boxed{x_i^{m0}}}^{\triangleright\triangleright \boxed{x_i^{m0}}} \triangleleft\triangleleft\ \overbrace{m \ \triangleright\triangleright \boxed{x_j^{m1}}}^{\triangleright\triangleright \boxed{x_j^{m1}}} \triangleleft\triangleleft\ e \quad \mathrlap{\overbrace{\phantom{x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}}}^{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}} x_i^{r1} x_i^{l1} \underbrace{x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1}}_{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}$$

In this scenario, the best that we are able to obtain is an overlap of two letters. This corresponds to the number of satisfied equations, namely $x_i \oplus x_{i+1} = 0$ and $x_j \oplus x_{j+1} = 0$.

CASE (ii) $\quad X_i \oplus X_j = 0$ **:**
Let us start with the constellation $(0X_{i+1}, 0X_{j+1})^s_{\{i,j\}}$. In this case, we set $\psi(x_i) = 0$ and $\psi(x_j) = 0$. Given the strings $\triangleright\triangleright \boxed{x_i^{m0}}$, $\boxed{X_{i+1}} \triangleleft\triangleleft$, $\triangleright\triangleright \boxed{x_j^{r0}}$ and $\boxed{X_{j+1}} \triangleleft\triangleleft$ with $X_{i+1} \in \{x_i^{m0}, x_i^{l1}\}$ and $X_{j+1} \in \{x_j^{m1}, x_j^{l0}\}$, we obtain the following scenario:

$$b \rhd\rhd \boxed{x_i^{m0}}\, x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}\, \boxed{X_{i+1}} \lhd\lhd\, m \rhd\rhd \overbrace{\boxed{x_j^{r0}}\, x_j^{l0}x_i^{r1}x_i^{l1}}^{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}} \boxed{X_{j+1}} \lhd\lhd\, e$$

$$\downarrow$$

$$b \rhd\rhd \boxed{x_i^{m0}}\,\boxed{X_{i+1}} \lhd\lhd\, m \rhd\rhd \overbrace{\boxed{x_j^{r0}}\, x_j^{l0}\, \underbrace{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}_{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}}^{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}} \boxed{X_{j+1}} \lhd\lhd\, e$$

The most advantageous simple alignment in this case is the 0-alignment of the strings in $S(g^x_{\{i,j\}})$. If $\psi(x_i) = \psi(x_{i+1}) = 0$ holds, which means $X_{i+1} = x_i^{m0}$, we obtain another overlap of one letter by aligning $\rhd\rhd \boxed{x_i^{m0}}$ with $\boxed{x_i^{m0}} \lhd\lhd$. A similar argument holds for $\psi(x_j) = \psi(x_{j+1}) = 0$. Notice that the equation $x_i \oplus x_j = 0$ is satisfied by ψ_s. In summary, we state that we obtain an overlap of one additional letter per satisfied equation. Hence, we obtain an overlap of three letters according to the satisfied equations $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 0$ and $x_j \oplus x_{j+1} = 0$.

Consider the constellation $(1X_{i+1}, 1X_{j+1})^s_{\{i,j\}}$. Hence, we are given the strings $\rhd\rhd \boxed{x_i^{r1}}$, $\boxed{X_{i+1}} \lhd\lhd$, $\rhd\rhd \boxed{x_j^{m1}}$ and $\boxed{X_{j+1}} \lhd\lhd$ with $X_{i+1} \in \{x_i^{m0}, x_i^{l1}\}$ and $X_{j+1} \in \{x_j^{m1}, x_j^{l0}\}$. We obtain the scenario displayed below.

$$b \rhd\rhd \boxed{x_i^{r1}}\, x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}\, \boxed{X_{i+1}} \lhd\lhd\, m \rhd\rhd \boxed{x_j^{m1}}\, x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}\, \boxed{X_{j+1}} \lhd\lhd\, e$$

$$\downarrow$$

$$b \rhd\rhd \overbrace{\boxed{x_i^{r1}}\, x_i^{l1}\, \underbrace{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}_{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}}^{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}} \boxed{X_{i+1}} \lhd\lhd\, m \rhd\rhd \boxed{x_j^{m1}}\,\boxed{X_{j+1}} \lhd\lhd\, e$$

In this case, we use the 1-alignment of the strings in $S(g^x_{\{i,j\}})$. If $\psi(x_i) = \psi(x_{i+1}) = 1$ holds, which means $X_{i+1} = x_i^{l1}$, we obtain another overlap of one letter by aligning

$$\rhd\rhd \overbrace{\boxed{x_i^{r1}}\, x_i^{l1}\, \underbrace{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}_{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}}^{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}} \quad \text{with} \quad \boxed{x_i^{l1}} \lhd\lhd\, .$$

In case of $\psi(x_j) = \psi(x_{j+1}) = 1$, we may apply a similar argument. Notice that the equation $x_i \oplus x_j = 0$ is satisfied by ψ_s. In summary, we state that we obtain an overlap of one additional letter per satisfied equation. Hence, we obtain an overlap of three letters according to the satisfied equations $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 0$ and $x_j \oplus x_{j+1} = 0$.

CASE (iii) $X_i \oplus X_j = 1$ **and** $X_i \neq X_{i+1}$**:**
Let us begin with the constellation $(10, 0X_{j+1})^s_{\{i,j\}}$. We consider the scenario depicted below, in which we are given the strings $\rhd\rhd \boxed{x_i^{r1}}$, $\boxed{x_i^{m0}} \lhd\lhd$, $\rhd\rhd \boxed{x_j^{l0}}$ and $\boxed{X_{j+1}} \lhd\lhd$ with $X_{j+1} \in \{x_j^{l0}, x_j^{m1}\}$.

$$b \rhd\rhd \boxed{x_i^{r1}}\, x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0} \boxed{x_i^{m0}} \lhd\lhd\ m \rhd\rhd \boxed{x_j^{r0}}\, x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1} \boxed{X_{j+1}} \lhd\lhd\ e$$

$$\downarrow$$

$$\overbrace{b \rhd\rhd \boxed{x_i^{m0}}}^{\rhd\rhd \boxed{x_i^{m0}}} \lhd\lhd\ m\ \rhd\rhd \mathrlap{\overbrace{\phantom{\boxed{x_j^{r0}}x_j^{l0}x_i^{r1}x_i^{l1}}}^{x_j^{r0}x_j^{l0}x_i^{r1}x_i^{l1}}}\boxed{x_j^{r0}}x_j^{l0}\underbrace{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}}_{x_i^{r1}x_i^{l1}x_j^{r0}x_j^{l0}} \boxed{X_{j+1}} \lhd\lhd\ e$$

Instead of using the 1-alignment of the strings in $S(g_i^x)$, we rather switch to the 0-alignment, i.e. we obtain the string $\rhd\rhd \boxed{x_i^{m0}}$ and define $\psi(x_i) = 0$. It results directly in gaining two additional satisfied equations and an overlap of one additional letter. As a matter of fact, we might lose an overlap of one letter, because the string $\rhd\rhd \boxed{x_1^{m}}$ might have been aligned from the right side with another string. Furthermore, the equation $x_{i-1} \oplus x_i = 0$ might be unsatisfied. But all in all, we obtain at least $2 - 1$ additional satisfied equations by switching the value without increasing the superstring. Notice that we may achieve an additional overlap of one letter if $X_{j+1} = x_j^{l0}$ holds, which means that ψ_s satisfies the equation $x_j \oplus x_{j+1} = 0$.

The next constellation we are going to analyze is $(01, 1X_{j+1})^s_{\{i,j\}}$. Hence, we are given the strings $\rhd\rhd \boxed{x_i^{m0}}$, $\boxed{x_i^{l1}}$ $\lhd\lhd$, $\rhd\rhd$ $\boxed{x_j^{m1}}$ and $\boxed{X_{j+1}}$ $\lhd\lhd$, with $X_{j+1} \in \{x_j^{l0}, x_j^{m1}\}$. The situation is displayed below.

$$
b \rhd\!\rhd \boxed{x_i^{m0}} x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0} \boxed{x_i^{l1}} \lhd\!\lhd\ m \rhd\!\rhd \boxed{x_j^{m1}} x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1} \boxed{X_{j+1}} \lhd\!\lhd\ e
$$

$$
\downarrow
$$

$$
b \rhd\!\rhd \overbrace{\boxed{x_i^{r1}} x_i^{l1} \underbrace{x_j^{r0} x_j^{l0} x_i^{r1} \boxed{x_i^{l1}}}_{x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1}}}^{x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}} \lhd\!\lhd\ m \rhd\!\rhd \boxed{x_j^{m1}} \boxed{X_{j+1}} \lhd\!\lhd\ e
$$

We obtain a similar situation, in which we switch $\rhd\!\rhd \boxed{x_i^{m0}}$ to $\rhd\!\rhd \boxed{x_i^{r1}}$. Accordingly, we define $\psi_s(x_i) = 1$.. We obtain at least one additional satisfied equation by switching the value without increasing the length of the superstring. Notice that we may achieve an additional overlap of one letter if $X_{j+1} = x_j^{m1}$ holds. It corresponds to the satisfied equation $x_j \oplus x_{j+1} = 0$.

CASE (iv) $X_i \oplus X_j = 1$**,** $X_j \neq X_{j+1}$ **and** $X_i = X_{i+1}$**:**
Starting our analysis with the constellation $(00, 10)^s_{\{i,j\}}$, we obtain the following scenario.

$$
b \rhd\!\rhd \boxed{x_i^{m0}} x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0} \boxed{x_i^{m0}} \lhd\!\lhd\ m \rhd\!\rhd \boxed{x_j^{m1}} x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1} \boxed{x_j^{l0}} \lhd\!\lhd\ e
$$

$$
\downarrow
$$

$$
b \rhd\!\rhd \overbrace{\boxed{x_i^{m0}}}^{\rhd\rhd \boxed{x_i^{m0}}} \lhd\!\lhd\ m \rhd\!\rhd \overbrace{\boxed{x_j^{r0}} x_j^{l0} \underbrace{x_i^{r1} x_i^{l1} x_j^{r0} \boxed{x_j^{l0}}}_{x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}}}^{x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1}} \lhd\!\lhd\ e
$$

In this case, we argue that we switch the string $\rhd\!\rhd \boxed{x_j^{m1}}$ to $\rhd\!\rhd \boxed{x_j^{r0}}$. This means that we set $\psi_s(x_j) = 0$. This transformation yields an overlap of at least the same number of letters, since we might lose an overlap of one letter from the left side. On the other hand, we align the string

$$
\rhd\!\rhd \boxed{x_j^{r0}} \quad \text{with} \quad \overbrace{x_j^{r0} x_j^{l0} \underbrace{x_i^{r1} x_i^{l1} x_j^{r0} \boxed{x_j^{l0}}}_{x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}}}^{x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1}} \lhd\!\lhd
$$

from the right side by one letter. Notice that we gain at least one additional satisfied equation.

The last constellation we are going to analyze is $(11, 01)^s_{\{i,j\}}$. The corresponding situation is depicted below.

$$b \rhd\rhd \boxed{x_i^{r1}}\, x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0} \boxed{x_i^{l1}} \lhd\lhd\, m \rhd\rhd \boxed{x_j^{r0}}\, x_j^{r0} x_j^{l0} x_i^{r1} x_i^{l1} \boxed{x_j^{m1}} \lhd\lhd\, e$$

$$\downarrow$$

$$b \rhd\rhd \overbrace{\boxed{x_i^{r1}}\, x_i^{l1} x_j^{r0} x_j^{l0}}^{x_i^{r1} x_i^{l1} x_j^{r0} x_j^{l0}} x_i^{r1} \boxed{x_i^{l1}} \lhd\lhd\, m \rhd\rhd \overbrace{\boxed{x_j^{m1}}}^{\rhd\rhd \boxed{x_j^{m1}}} \lhd\lhd\, e$$

In this case, we switch the string $\rhd\rhd \boxed{x_j^{r0}}$ to $\rhd\rhd \boxed{x_j^{m1}}$. Similarly to the former case, this transformation does not increase the length of the superstring. By defining $\psi_s(x_j) = 1$, we achieve at least one more satisfied equation.

In summary, we note that we achieve at least the same number of satisfied equations as the number of overlapped letters. By applying the defined transformation, the superstring contains only strings corresponding to matching equations, which use a simple alignment.

In case of matching equations $x_i \oplus x_j = 0$ with $i > j$ can be analyzed analogously. We are going to define the assignment for contact variables.

Defining the Assignment for Contact Variables

Let $g_j^3 \equiv x \oplus y \oplus z = 0$ be an equation with exactly three variables in $\mathcal{H}$. Given a simple alignment of the strings corresponding to the equations $x_{j_1-1} \oplus x = 0$, $x \oplus x_{j_1+1} = 0$, $y_{j_2-1} \oplus y = 0$, $y \oplus y_{j_2+1} = 0$, $z_{j_3-1} \oplus z = 0$, and $z \oplus z_{j_3+1} = 0$, we are going to define an assignment based on the underlying simple alignments and analyze the number of satisfied equations in dependence of the number of overlapped letters in the superstring.

For a given superstring s for $\mathcal{S}_\mathcal{H}$ and equation $g_j^3 \equiv x \oplus y \oplus z = 0$, we define a *constellation* c given by $(X_1X_2, Y_1Y_2, Z_1Z_2)_j^s$ with $X_1, X_2, Y_1, Y_2, Z_1, Z_2 \in \{0, 1\}$, whereby $C = 1$ if and only if the strings in the corresponding set are using a 1-alignment in s. A constellation denotes which of the simple alignments is used by the strings in s. We call a constellation inconsistent if there is an entry A_1A_2 such that $A_1 \neq A_2$. Otherwise, c is called consistent.

Based on a constellation for a given superstring and an equation g_j^3 with three variables, we are going to define the assignment ψ_s for the variables in g_j^3.

Definition 7 (Assignment ψ_s to Contact Variables)**.** *Let $\mathcal{H}$ be an instance of the Hybrid problem, $\mathcal{S}_\mathcal{H}$ its corresponding instance of the superstring problem, s a super-*

string for $\mathcal{S}_{\mathcal{H}}$ and $g_j^3 \equiv x \oplus y \oplus z = 0$ an equation with three variables in $\mathcal{H}$. For the associated constellation $c = (X_1X_2, Y_1Y_2, Z_1Z_2)_j^s$, we define ψ_s in the following way.

(i) *If c is consistent, then, we define $\psi_s(x) = X_1$, $\psi_s(y) = Y_1$ and $\psi_s(z) = Z_1$*

(ii) *Otherwise, let A_1A_2 be an entry in c with $A_1 \neq A_2$ and α its corresponding variable. Furthermore, let β and γ be variables associated with the entry B_1B_2 and C_1C_2, respectively. If $A_1 \oplus B_1 \oplus C_1 = 0$ holds, we define $\psi_s(\alpha) = A_1$, $\psi_s(\beta) = B_1$ and $\psi_s(\gamma) = C_1$.*

(iii) *Otherwise, we have $A_1 \oplus B_1 \oplus C_1 = 1$. Then, we define $\psi_s(\alpha) = 1 - A_1$, $\psi_s(\beta) = B_1$ and $\psi_s(\gamma) = C_1$.*

We are going to analyze the following three cases and define the transformations for the actual superstring for $\mathcal{S}_{\mathcal{H}}$.

(i) $X_1 \oplus Y_1 \oplus Z_1 = 1$ and c is consistent

(ii) $X_1 \oplus Y_1 \oplus Z_1 = 0$ and c is inconsistent

(iii) $X_1 \oplus Y_1 \oplus Z_1 = 1$ and c is inconsistent

Let us begin with case (i).

CASE (i) $X_1 \oplus Y_1 \oplus Z_1 = 1$ **and** c **is consistent:**
In this case, we start with the constellation $(11, 11, 11)_j^s$. We depict the considered situation below.

$$b \triangleright\triangleright \boxed{x^{r1}}\boxed{x^{l1}} \triangleleft\triangleleft\ m_1 \triangleright\triangleright \boxed{y^{r1}}\, y^{r1}A_jy^{l1}\boxed{y^{l1}} \triangleleft\triangleleft\ m_2 \triangleright\triangleright \boxed{z^{r1}}\, z^{r1}B_jz^{l1}\boxed{z^{l1}} \triangleleft\triangleleft\ e$$

$$\downarrow$$

$$b \triangleright\triangleright \boxed{x^{r1}}\boxed{x^{l1}} \triangleleft\triangleleft\ m_1 \triangleright\triangleright \underbrace{\boxed{y^{r1}}A_j\boxed{y^{l1}}}_{y^{r1}Ay^{l1}} \triangleleft\triangleleft\ m_2 \triangleright\triangleright \underbrace{\boxed{z^{r1}}B_j\boxed{z^{l1}}}_{z^{r1}Bz^{l1}} \triangleleft\triangleleft\ e$$

According to the definition of ψ_s, we have $\psi(x) = \psi(y) = \psi(z) = 1$. Notice that the equation $x \oplus y \oplus z = 0$ is unsatisfied. On the other hand, the assignment ψ_s satisfies the equations $x \oplus x_{j_1+1} = 0$, $y \oplus y_{j_2+1} = 0$ and $z \oplus z_{j_3+1} = 0$.
We note that a string corresponding to $S^A(g_j^3)$ or $S^B(g_j^3)$ using a simple alignment can have an overlap of at most one letter from the right side as well as from the left side. Therefore, the best we can hope for is to overlap the string $y^{r1}Ay^{l1}$ with $\triangleright\triangleright \boxed{y^{r1}}$ and $\boxed{y^{l1}} \triangleleft\triangleleft$ by one letter in each case. The same holds for the string $z^{r1}B_jz^{l1}$. Consequently, we conclude that the number of overlapped letters is bounded from above by four.

In case of $X_1 + Y_1 + Z_1 = 1$, we analyze exemplary the constellation $(00, 00, 11)^s_j$. We set $\psi(z) = 1$, $\psi(x) = 0$ and $\psi(y) = 0$. This situation is displayed below.

$$
\begin{array}{c}
b \rhd\rhd \boxed{x^{m0}}\boxed{x^{m0}} \lhd\lhd m_1 \rhd\rhd \boxed{y^{m0}} y^{r1}A_j y^{l1} \boxed{y^{m0}} \lhd\lhd m_2 \rhd\rhd \boxed{z^{r1}} z^{r1}B_j z^{l1} \boxed{z^{l1}} \lhd\lhd e \\
\downarrow \\
b \underbrace{\rhd\rhd \boxed{x^{m0}}}_{\rhd\rhd \boxed{x^{m0}}} \lhd\lhd m_1 \underbrace{\rhd\rhd \boxed{y^{m0}}}_{\rhd\rhd \boxed{y^{m0}}} \lhd\lhd m_2 \rhd\rhd \underbrace{\boxed{z^{r1}} B_j \boxed{z^{l1}}}_{z^{r1}B_j z^{l1}} \lhd\lhd e\, y^{r1}A_j y^{l1}
\end{array}
$$

Due to the z^1-alignment of the strings in $S^B(g^3_j)$, we obtain an overlap of two letters. Additionally, we align the string $\rhd\rhd \boxed{x^{m0}}$ from the left with $\boxed{x^{m0}} \lhd\lhd$. The same holds for $\rhd\rhd \boxed{y^{m0}}$ and $\boxed{y^{m0}} \lhd\lhd$. Notice that it is not more advantageous to align the string $x^{m0}B_jC_j$ with $\rhd\rhd \boxed{x^{m0}}$, since we lose the overlap of one letter with $\boxed{x^{m0}} \lhd\lhd$. Hence, we are able to get an overlap of at most four letters, which corresponds to the satisfied equations $x \oplus x_{j_1+1} = 0$, $y \oplus y_{j_2+1} = 0$ and $z \oplus z_{j_3+1} = 0$.

CASE $X_1 \oplus Y_1 \oplus Z_1 = 0$ and c is inconsistent:
First, we concentrate on the constellations with the property $X_1 + Y_1 + Z_1 = 2$. Exemplary, we analyze the constellation $(0X_2, 1Y_2, 1Z_2)^s_j$ depicted below.

$$
\begin{array}{c}
b \rhd\rhd \boxed{x^{m0}}\boxed{X_2} \lhd\lhd m_1 \rhd\rhd \boxed{y^{r1}} y^{r1}A_j y^{l1} \boxed{Y_2} \lhd\lhd m_2 \rhd\rhd \boxed{z^{r1}} z^{r1}B_j z^{l1} \boxed{Z_2} \lhd\lhd e \\
\downarrow \\
b \rhd\rhd \boxed{x^{m0}}\boxed{X_2} \lhd\lhd m_1 \rhd\rhd \underbrace{\boxed{y^{r1}} A_j y^{l1}}_{y^{r1}A_j y^{l1}} \boxed{Y_2} \lhd\lhd m_2 \rhd\rhd \underbrace{\boxed{z^{r1}} B_j\, z^{l1}}_{z^{r1}B_j z^{l1}} \boxed{Z_2} \lhd\lhd e
\end{array}
$$

The strings $\rhd\rhd \boxed{y^{r1}}$ and $\rhd\rhd \boxed{z^{r1}}$ can be used to align from the right side with $z^{r1}Bz^{l1}$ and $y^{r1}Ay^{l1}$, respectively. It yields an overlap of two letters. If the corresponding equations with two variables are satisfied, which means $X_2 = x^{m0}$, $Y_2 = y^{l1}$ and $Z_2 = z^{l1}$, we gain an overlap of one letter per satisfied equation. Notice that using the x^0-alignment of $S(g^3_j)$ does not yield more overlapped letters. In summary, it is possible to attain an overlap of at most five letters, which corresponds to the constellation $(00, 11, 11)^s_j$. An analogue argumentation holds for the constellations $(1X_2, 1Y_2, 0Z_2)^s_j$ and $(1X_2, 0Y_2, 1Z_2)^s_j$.

Next, we discuss constellations with the property $X_1 + Y_1 + Z_1 = 0$. For this reason, we consider the constellation $(0X_2, 0Y_2, 0Z_2)^s_j$.

$$b \rhd\rhd \boxed{x^{m0}}\boxed{X_2} \lhd\lhd m_1 \rhd\rhd \boxed{y^{m0}} y^{r1}B_jy^{r1}\boxed{Y_2} \lhd\lhd m_2 \rhd\rhd \boxed{z^{m0}} y^{r1}B_jy^{r1}\boxed{Z_2} \lhd\lhd e$$

$$\downarrow$$

$$b \rhd\rhd \underbrace{\boxed{x^{m0}}A\overbrace{C_j}^{C_jBx^{m0}}}_{x^{m0}AC_j}\!\!Bx^{m0}\boxed{X_2} \lhd\lhd m_1 \rhd\rhd \boxed{y^{m0}}\boxed{Y_2} \lhd\lhd m_2 \rhd\rhd \boxed{z^{m0}}\boxed{Z_2} \lhd\lhd e$$

Recall that $x^{m0}C_jx^{m0}$ denotes the x^0-alignment of $S(g^3_j)$. This string can be aligned from the left with $\rhd\rhd\ x^{m0}$. If $X_2 = x^{m0}$ holds, we achieve another overlap of one letter. Furthermore, the string $\rhd\rhd \boxed{y^{m0}}$ can be aligned from the right with $\boxed{Y_2} \lhd\lhd$ if and only if $Y_2 = y^{m0}$ holds. A similar argumentation can be applied to the strings $\rhd\rhd \boxed{z^{m0}}$ and $\boxed{Z_2} \lhd\lhd$. Finally, we note that we cannot benefit by aligning the string $\boxed{y^{l1}} \lhd\lhd$ with $y^{r1}Ay^{l1}$. Consequently, we see that using the string $x^{m0}C_jx^{m0}$ is generally more profitable. All in all, we gain an additional overlap of one letter for satisfying $x \oplus y \oplus z = 0$ and another overlap of one letter if the equation with two variables corresponding to the considered variable is satisfied.

CASE $X_1 \oplus Y_1 \oplus Z_1 = 1$ and c is inconsistent:
Let us start with constellations satisfying $X_1 + Y_1 + Z_1 = 3$. Exemplary, we analyze the constellation $(10, 1Y_2, 1Z_2)^s_j$. Due to the definition of ψ_s, we set $\psi(x) = 1 - X_1$, $\psi(y) = 1$ and $\psi(z) = 1$. Notice that ψ_s satisfies the equation $x \oplus y \oplus z = 0$. By switching the value $\psi_s(x)$ from X_1 to $1 - X_1$, the equation $x_{j_1-1} \oplus x = 0$ might become unsatisfied. Furthermore, we might lose an overlap of one letter by flipping the 1-alignment of the strings corresponding to $x_{j_1-1} \oplus x = 0$ to the 0-alignment. On the other hand, we gain an overlap of one letter by aligning the string $\rhd\rhd \boxed{x^{m0}}$ from the right side with $\boxed{x^{m0}} \lhd\lhd$. This transformation yields at least one more satisfied equation. In addition, the strings $y^{r1}A_jy^{l1}$ and $z^{r1}Bz^{l1}$ can be aligned by one letter with $\rhd\rhd \boxed{y^{r1}}$ and $\rhd\rhd \boxed{z^{r1}}$, respectively. If $Z_2 = z^{l1}$ and $Y_2 = y^{l1}$ holds, we achieve another overlap of one letter in each case. The situation is depicted below.

$$b \rhd\rhd \boxed{x^{r1}}\,\boxed{x^{m0}} \lhd\lhd m_1 \rhd\rhd \boxed{y^{r1}}\, y^{r1} A y^{l1} \boxed{Y_2} \lhd\lhd m_2 \rhd\rhd \boxed{z^{r1}}\, z^{r1} B z^{l1} \boxed{Z_2} \lhd\lhd e$$

$$\downarrow$$

$$\underbrace{b \rhd\rhd \boxed{x^{m0}}}_{\rhd\rhd \boxed{x^{m0}}} \lhd\lhd\rhd\rhd \underbrace{\boxed{y^{r1}} A_j y^{l1}}_{y^{r1} A_j y^{l1}} \boxed{Y_2} \lhd\lhd\rhd\rhd \underbrace{\boxed{z^{r1}} B_j z^{l1}}_{z^{r1} B_j z^{l1}} \boxed{Z_2} \lhd\lhd$$

The other constellations satisfying $X_1 + Y_1 + Z_1 = 3$ can be analyzed analogously.

The remaining constellations $(X_1X_2, Y_1Y_2, Z_1Z_2)^s_j$ to be discussed satisfy $X_1 + Y_1 + Z_1 = 1$ and are inconsistent. Exemplary, we analyze the constellation $(01, 0Y_2, 1Z_2)^s_j$. For $(01, 0Y_2, 1Z_2)^s_j$, we set $\psi(x) = 1 - X_1$, $\psi(y) = Y_1$ and $\psi(z) = Z_1$. The scenario is depicted below.

$$b \rhd\rhd \boxed{x^{m0}}\, x^{r1} A_j x^{l1} \boxed{x^{l1}} \lhd\lhd m_1 \rhd\rhd \boxed{y^{m0}}\,\boxed{Y_2} \lhd\lhd m_2 \rhd\rhd \underbrace{\boxed{z^{r1}} B_j z^{l1}}_{z^{r1} B_j z^{l1}} \boxed{Z_2} \lhd\lhd e$$

$$\downarrow$$

$$b \rhd\rhd \underbrace{\boxed{x^{r1}} A_j \boxed{x^{l1}}}_{x^{r1} A_j x^{l1}} \lhd\lhd m_1 \rhd\rhd \boxed{y^{m0}}\,\boxed{Y_2} \lhd\lhd m_2 \rhd\rhd \underbrace{\boxed{z^{r1}} B z^{l1}}_{z^{r1} B z^{l1}} \boxed{Z_2} \lhd\lhd e$$

By flipping the 0-alignment of the strings corresponding to $x_{j_1-1} \oplus x = 0$ to the 1-alignment, we can overlap $x^{r1}A_jx^{l1}$ from the left side with $\rhd\rhd \boxed{x^{r1}}$ and with $\boxed{x^{l1}} \lhd\lhd$ from the right side. This transformation achieves an overlap of at most one more letter. Moreover, we obtain at least one more satisfied equation by using this definition of ψ_s. If $Z_2 = z^{l1}$ and $Y_2 = y^{m0}$ holds, it yields an overlap of three additional letters, which corresponds to the constellation $(11, 00, 11)^s_j$.

In summary, we note that it is possible to achieve an overlap of at least one letter in each case. In addition to it, the assignment ψ_s yields at least the same number of satisfied equations as the number of overlapped letters, which can be achieved due to the transformations. This means that if ψ_s satisfies the equations g^3_j, $x \oplus x_{j_1+1} = 0$, $y \oplus y_{j_2+1} = 0$ and $z \oplus z_{j_3+1} = 0$, the corresponding strings in s can have an overlap of at most five letters.

5.5 Proof of Theorem 6

Given an instance $\mathcal{H}$ of the Hybrid problem with n circles, m_2 equations with two variables and m_3 equations with exactly three variables with the properties described in Theorem 5, we construct in polynomial time an instance $\mathcal{S}_{\mathcal{H}}$ of the Shortest Superstring problem with the properties described in section 5.2. Let ϕ be an assignment to the variables of $\mathcal{H}$ which leaves at most u equations unsatisfied. According to section 5.3, the length of the superstring s_ϕ is at most

$$|s_\phi| \leq 7 \cdot n + 5 \cdot m_2 + 22 \cdot m_3 + u,$$

since the length of the superstring increases by at most one letter for every unsatisfied equation of the assignment. Regarding the compression measure, we obtain the following.

$$\begin{aligned} comp(\mathcal{S}_{\mathcal{H}}, s_\phi) &\geq \sum_{s \in \mathcal{S}_{\mathcal{H}}} |s| - (7 \cdot n + 5 \cdot m_2 + 22 \cdot m_3 + u) \\ &= (4+8)n + 8 \cdot m_2 + 36 \cdot m_3 - (7 \cdot n + 5 \cdot m_2 + 22 \cdot m_3 + u) \\ &= 5n + 3m_2 + 14m_3 - u \end{aligned}$$

On the other hand, given an superstring s for $S_{\mathcal{H}}$ with length

$$|s| = 5m_2 + 22m_3 + u + 7n,$$

we can construct in polynomial time an normed superstring s' without increasing the length of it by applying the transformations defined in section 5.4. This enables us to define an assignment ψ_s to the variables of $\mathcal{H}$ according section 5.4 that leaves at most u equations in $\mathcal{H}$ unsatisfied. A similar argumentation leads to the conclusion that given a superstring s for $S_{\mathcal{H}}$ with compression

$$comp(\mathcal{S}_{\mathcal{H}}, s_\phi) = 5n + 3m_2 + 14m_3 - u,$$

we construct in polynomial time an assignment to the variables in $\mathcal{H}$ such that at most u equations are unsatisfied. $\square$

6 Concluding Remarks

It seems that a new method is needed now in order to obtain better approximation lower bounds. Perhaps direct PCP constructions are the natural next step for proving stronger approximation hardness results for the problems considered in this paper.

References

[AS95] C. Armen and C. Stein, *Improved Length Bounds for the Shortest Superstring Problem*, Proc. 5th WADS (1995), LNCS 955, 1995, pp. 494–505.

[AS98] C. Armen, C. Stein, *A $2\frac{2}{3}$ Superstring Approximation Algorithm*, Discrete Applied Mathematics 88 (1998), pp. 29–57.

[B02] M. Bläser, *An $\frac{8}{13}$ Approximation Algorithm for the Asymmetric Max-TSP*, Proc. 13th SODA (2002), 64–73.

[BK99] P. Berman, M. Karpinski, *On Some Tighter Inapproximability Results*, Proc. 26th ICALP (1999), LNCS 1644, 1999, pp. 200–209.

[BJJ97] D. Breslauer, T. Jiang, and Z. Jiang, *Rotations of Periodic Strings and Short Superstrings*, J. Algorithms 24 (1997), pp. 340–353.

[BJL+94] A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yanakakis, *Linear Approximation of Shortest Superstrings*, J. ACM 41 (1994), pp. 630–647.

[CGP+94] A. Czumaj, L. Gasieniec, M. Piotrow, and W. Rytter, *Parallel and Sequential Approximations of Shortest Superstrings*, Proc. 1st SWAT (1994), LNCS 824, 1994, pp. 95–106.

[E99] L. Engebretsen, *An Explicit Lower Bound for TSP with Distances One and Two*, Algorithmica 35 (2003), pp. 301–318.

[EK06] L. Engebretsen and M. Karpinski, *TSP with Bounded Metrics*, J. Comput. Syst. Sci. 72 (2006), pp. 509–546.

[FNW79] M.L. Fisher, G.L. Nemhauser and L.A. Wolsey, *An Analysis of Approximations for Finding a Maximum Weight Hamiltonian Circuit*, Oper. Res. 27 (1979), pp. 799-809.

[GMS80] J. Gallant, D. Maier and J. A. Storer, *On Finding Minimal Length Superstrings* J. Comp. Sys. Sci. 20 (1980), pp. 50–58.

[H01] J. Håstad, *Some Optimal Inapproximability Results*, J. ACM 48 (2001), pp. 798–859.

[KLS+05] H. Kaplan, M. Lewenstein, N. Shafrir and M. Sviridenko, *Approximation Algorithms for Asymmetric TSP by Decomposing Directed Regular Multigraphs*, J. ACM 52 (2005), pp. 602–626.

[KPS94] R. Kosaraju, J. Park, and C. Stein, *Long Tours and Short Superstrings*, Proc. 35th FOCS (1994), pp. 166–177.

[L88] *Computational Molecular Biology, Sources and Methods for Sequence Analysis*, A. Lesk, Oxford University Press, 1988.

[L90] *Towards a DNA Sequencing Theory*, M. Li, Proc. 31th FOCS (1990), pp. 125–134.

[LS03] M. Lewenstein and M. Sviridenko, *Approximating Asymmetric Maximum TSP*, Proc. 14th SODA (2003), pp. 646–654.

[M94] M. Middendorf, *More on the Complexity of Common Superstring and Supersequence Problems*, Theor. Comput. Sci. 125 (1994), pp. 205–228.

[M98] M. Middendorf, *Shortest Common Superstrings and Scheduling with Coordinated Starting Times*, Theor. Comput. Sci. 191 (1998), pp. 205–214.

[MJ75] A. Mayne and E. B. James, *Information Compression by Factorising Common Superstrings*, Comput. J. 18 (1975), pp. 157–160.

[MS77] D. Maier and J.A. Storer, *A Note on the Complexity of the Superstring Problem*, Report No. 223, Computer Science Laboratory, Princeton University, 1977.

[O99] S. Ott, *Lower Bounds for Approximating Shortest Superstrings over an Alphabet of Size* 2, Proc. 25th. WG (1999), LNCS 1665, 1999, pp. 55-64.

[PY93] C. H. Papadimitriou and M. Yannakakis, *The traveling Salesman Problem with Distances One and Two*, Math. Oper. Res. 18 (1993), pp. 1–11.

[S88] J. A. Storer, *Data Compression: Methods and Theory*, Computer Science Press, 1988.

[S99] Z. Sweedyk, *A* $2\frac{1}{2}$*-Approximation Algorithm for Shortest Superstring*, SIAM J. Comput. 29 (1999), pp. 954–986.

[SS82] J. A. Storer and T. G. Szymanski, *Data Compression via Textual Substitution*, J. ACM 29 (1982), pp. 928–951.

[T90] V. G. Timkovskii, *Complexity of Common Subsequence and Supersequence Problems and Related Problems*, Cybernetics and Systems Analysis 25 (1990), pp. 565–580; translated from Kibernetika 25 (1989), pp. 1–13.

[T89] J. Turner, *Approximation Algorithms for the Shortest Common Superstring problem* , Information and Computation 83 (1989), pp. 1–20.

[TT93] S. Teng and F. Yao, *Approximating shortest superstrings*, Proc. 34th FOCS (1993), pp. 158–165, 1993.

[TU88] J. Tarhio and E. Ukkonen, *A Greedy Approximation Algorithm for Constructing Shortest Common Superstrings*, Theor. Comput. Sci. 57 (1988), pp. 131–145.

[V05] V. Vassilevska, *Explicit Inapproximability Bounds for the Shortest Superstring Problem*, Proc. 30th MFCS (2005), LNCS 3618, 2005, pp. 793–800.

[V92] S. Vishwanathan, *An Approximation Algorithm for the Asymmetric Travelling Salesman Problem with Distances One and Two*, Inf. Process. Lett. 44 (1992), pp. 297–302.