Improved Approximation for the Maximum Duo-Preservation String Mapping Problem*

Nicolas Boria, Adam Kurpisz, Samuli Leppänen, and Monaldo Mastrolilli

Dalle Molle Institute for Artificial Intelligence (IDSIA), Manno, Switzerland

Abstract. In this paper we present improved approximation results for the MAX DUO-PRESERVATION STRING MAPPING problem (MPSM) introduced in [Chen et al., Theoretical Computer Science, 2014] that is complementary to the well-studied MIN COMMON STRING PARTITION problem (MCSP). When each letter occurs at most k times in each string the problem is denoted by k-MPSM. First, we prove that k-MPSM is APX-Hard even when $k = 2$. Then, we improve on the previous results by devising two distinct algorithms: the first ensures approximation ratio 8/5 for $k = 2$ and ratio 3 for $k = 3$, while the second guarantees approximation ratio 4 for any bigger value of k. Finally, we address the approximation of CONSTRAINED MAXIMUM INDUCED SUBGRAPH (CMIS, a generalization of MPSM, also introduced in [Chen et al., Theoretical Computer Science, 2014]), and improve the best known 9-approximation for 3-CMIS to a 6-approximation, by using a configuration LP to get a better linear relaxation. We also prove that such a linear program has an integrality gap of k, which suggests that no constant approximation (i.e. independent of k) can be achieved through rounding techniques.

Keywords: Polynomial approximation, Max Duo-Preserving String Mapping Problem, Min Common String Partition Problem, Linear Programming, Configuration LP.

1 Introduction

String comparison is a central problem in stringology with a wide range of applications, including data compression, and bio-informatics. There are various ways to measure the similarity of two strings: one may use the Hamming distance which counts the number of positions at which the corresponding symbols are different, the Jaro-Winkler distance, the overlap coefficient, etc. However in computer science, the most common measure is the so called *edit distance* that measures the minimum number of edit operations that must be performed to transform the first string into the second. In biology, this number may provide some measure of the kinship between different species based on the similarities of their DNA. In data compression, it may help to store efficiently a set of similar

* Research supported by the Swiss National Science Foundation project 200020_144491/1 “Approximation Algorithms for Machine Scheduling Through Theory and Experiments”, and by the Sciex-Project 12.311

D. Brown and B. Morgenstern (Eds.): WABI 2014, LNBI 8701, pp. 14–25, 2014.

yet different data (e.g. different versions of the same object) by storing only one "base" element of the set, and then storing the series of edit operations that result in the other versions of the base element.

The concept of edit distance changes definition based on the set of edit operations that are allowed. When the only edit operation that is allowed is to shift a block of characters, the edit distance can be measured by solving the MIN COMMON STRING PARTITION problem.

The MIN COMMON STRING PARTITION (MCSP) is a fundamental problem in the field of string comparison [7,13], and can be applied more specifically to genome rearrangement issues, as shown in [7]. Consider two strings A and B, both of length n, such that B is a permutation of A. Also, let $\mathcal{P}_A$ denote a *partition* of A, that is, a set of substrings whose concatenation results in A. The MCSP Problem introduced in [13] and [19] asks for a partition $\mathcal{P}_A$ of A and $\mathcal{P}_B$ of B of minimum cardinality such that $\mathcal{P}_A$ is a permutation of $\mathcal{P}_B$. The $k-$MCSP denotes the restricted version of the problem where each letters has at most k occurrences. This problem is NP-Hard and even APX-Hard, also when the number of occurrences of each letter is at most 2 (note that the problem is trivial when this number is at most 1) [13]. Since then, the problem has been intensively studied, especially in terms of polynomial approximation [7,8,9,13,15,16], but also parametric computation [4,17,10,14]. The best approximations known so far are an $O(\log n \log^* n)$-approximation for the general version of the problem [9], and an $O(k)$-approximation for $k-$MCSP [16]. On the other hand, the problem was proved to be Fixed Parameter Tractable (FPT), first with respect to both k and the cardinality ϕ of an optimal partition [4,10,14], and more recently, with respect to ϕ only [17].

In [6], the maximization version of the problem is introduced and denoted by MAX DUO-PRESERVATION STRING MAPPING (MPSM). Reminding that a *duo* denotes a couple of consecutive letters it is clear that when a solution $(\mathcal{P}_A, \mathcal{P}_B)$ for MIN COMMON STRING PARTITION partitions A and B into ϕ substrings, this solution can be translated as a mapping π from A to B that preserves exactly $n - \phi$ duos. Hence, given two strings A and B, the MPSM problem asks for a mapping π from A to B that preserves a maximum number of duos (a formal definition is given in Subsection 3.1). An example is provided in Figure 1.

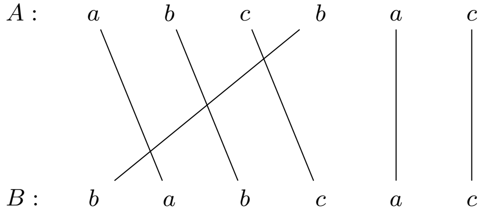


Fig. 1. A mapping π that preserves 3 duos

Considering that MCSP is NP-Hard [13], its maximization counterpart MPSM is also NP-Hard. However, these two problems might have different behaviors in terms of approximation, inapproximability, and parameterized complexity. MAX INDEPENDENT SET and MIN VERTEX COVER provide a good example of how symmetrical problems might have different characteristics: on the one hand, MAX INDEPENDENT SET is inapproximable within ratio $n^{\varepsilon-1}$ for a given $\varepsilon \in (0,1)$ unless $\mathbf{P} = \mathbf{NP}$ [18], and is $W[1]$-Hard [11]; and on the other hand MIN VERTEX COVER is easily 2-approximable in polynomial time by taking all endpoints of a maximal matching [12], and is FPT [5].

The authors of [6] provide some approximation results for MPSM in the following way: a graph problem called CONSTRAINED MAXIMUM INDUCED SUBGRAPH (CMIS) is defined and proved to be a generalization of MPSM. Using a solution to the linear relaxation of CMIS, it is proved that a randomized rounding provides a k^2 expected approximation ratio for k-CMIS (and thus for k-MPSM), and a 2 expected approximation ratio for 2-CMIS (and thus for 2-MPSM).

In what follows, we start by proving briefly that k-MPSM is APX-Hard, even when $k = 2$ (Section 2). Then, we present some improved approximation results for MPSM (Section 3), namely a general approximation algorithm that guarantees approximation ratio 4 regardless of the value of k (Subsection 3.2), and an algorithm that improves on this ratio for small values of k (Subsection 3.3). Finally, we improve on the approximation of 3-CMIS, by using a configuration LP to get a better relaxed solution (Section 4), and analyze the integrality gap of this relaxed solution.

2 Hardness of Approximation

We will show that MPSM is APX–hard, which essentially rules out any polynomial time approximation schemes unless P = NP. The result follows with slight modifications from the known approximation hardness result for MCSP. Indeed, in [13] it is shown that any instance of MAX INDEPENDENT SET in a cubic graph (3–MIS) can be reduced to an instance of 2–MCSP (proof of Theorem 2.1 in [13]). We observe that the construction used in their reduction also works as a reduction from 3–MIS to 2–MPSM. In particular, given a cubic graph with n vertices and independence number α, the corresponding reduction to 2–MPSM has an optimum value of $m = 4n + \alpha$.

Given a ρ–approximation to 2–MPSM, we will hence always find an independent set of size at least $\rho m - 4n$. It is shown in [3] that it is NP–hard to approximate 3–MIS within $\frac{139}{140} + \epsilon$ for any $\epsilon > 0$. Therefore, unless P = NP, for every $\epsilon > 0$ there is an instance I of 3–MIS such that:

$$\frac{\mathrm{APP}_I}{\mathrm{OPT}_I} \leqslant \frac{139}{140} + \epsilon$$

where APP_I is the solution produced by any polynomial time approximation algorithm and OPT_I the optimum value of I. Substituting here we get:

$$\frac{\rho m - 4n}{m - 4n} \leqslant \frac{139}{140} + \epsilon$$

Solving for ρ yields:

$$\rho \leqslant \frac{139}{140} + \frac{4n}{m}\left(\frac{1}{140} - \epsilon\right) + \epsilon \leqslant \frac{139}{140} + \frac{16}{17 \cdot 140} + \frac{1}{17}\epsilon$$

where the last inequality follows from noting that for any cubic graph the maximum independent set α is always at least of size $\frac{1}{4}n$.

3 Approximation Algorithms for MAX DUO-PRESERVATION STRING MAPPING

In this section we present two different approximation algorithms. First, a simple algorithm that provides a 4-approximation ratio for the general version of the problem, and then an algorithm that improves on this ratio for small values of k.

3.1 Preliminaries

For $i = 1, ..., n$, we denote by a_i the ith character of string A, and by b_i the ith character in B. We also denote by $D^A = (D_1^A, ..., D_{n-1}^A)$ and $D^B = (D_1^B, ..., D_{n-1}^B)$ the set of duos of A and B respectively. For $i = 1, ..., n-1$, D_i^A corresponds to the duo (a_i, a_{i+1}), and D_i^B corresponds to the duo (b_i, b_{i+1}).

A mapping π from A to B is said to be *proper* if it is bijective, and if, $\forall i = 1, ..., n$, $a_i = b_{\pi(i)}$. In other words, each letter of the alphabet in A must be mapped to the same letter in B for the mapping to be proper. A couple of duos $\left(D_i^A, D_j^B\right)$ is said to be *preservable* if $a_i = b_j$ and $a_{i+1} = b_{j+1}$. Given a mapping π, a preservable couple of duos $\left(D_i^A, D_j^B\right)$ is said to be *preserved by* π if $\pi(i) = j$ and $\pi(i+1) = j+1$. Finally, two preservable couples of duos $\left(D_i^A, D_j^B\right)$ and $\left(D_h^A, D_l^B\right)$ will be called *conflicting* if there is no proper mapping that preserves both of them. These conflicts can be of two types, w.l.o.g., we suppose that $i \leqslant h$ (resp. $j \leqslant l$):

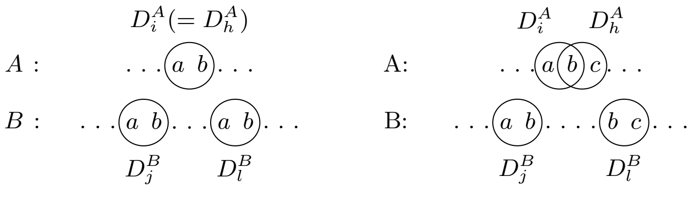
- Type 1: $i = h$ (resp. $j = l$) and $j \neq l$ (resp. $i \neq h$) (see Figure 2(a))
- Type 2: $i = h - 1$ (resp. $j = l - 1$) and $j \neq l - 1$ (resp. $i \neq h - 1$) (see Figure 2(b))

Let us now define formally the problem at hand:

Definition 1. MAX DUO-PRESERVATION STRING MAPPING *(MPSM):*

- ***Instance:*** *two strings A and B such that B is a permutation of A.*
- ***Solution:*** *a proper mapping π from A to B.*
- ***Objective:*** *maximizing the number of duos preserved by π, denoted by $f(\pi)$.*

Let us finally introduce the concept of *duo-mapping.* A duo-mapping σ is a mapping, which - unlike a mapping π that maps each character in A to a character in B - maps a *subset* of duos of D^A to a *subset* of duos of D^B. Having


$D_i^A (= D_h^A)$
A :
$\ldots (a\ b) \ldots$
B :
$\ldots (a\ b) \ldots (a\ b) \ldots$
D_j^B
D_l^B
D_i^A
D_h^A
A:
$\ldots (a (b) c) \ldots$
B:
$\ldots (a\ b) \ldots \ldots (b\ c) \ldots$
D_j^B
D_l^B

(a) Type 1

(b) Type 2

Fig. 2. Different types of conflicting pairs of duos

$\sigma(i) = j$ means that the duo D_i^A is mapped to the duo D_j^B. Again, a duo-mapping σ is said to be proper if it is bijective, and if $D_i^A = D_{\sigma(i)}^B$ for all duos mapped through σ. Note that a proper duo-mapping might map some conflicting couple of duos. Revisit the example of Figure 2(b): having $\sigma(i) = j$ and $\sigma(h) = l$ defines a proper duo-mapping that maps conflicting couple of duos. Notice however that a proper duo-mapping might generate conflicts of Type 2 only. We finally define the concept of *unconflicting* duo-mapping, which is a proper duo-mapping that does not map any pair of conflicting duos.

Remark 1. An unconflicting duo-mapping σ on some subset of duos of size $f(\sigma)$ immediatly derives a proper mapping π on the whole set of characters with $f(\pi) \geqslant f(\sigma)$: it suffices to map characters mapped by σ in the same way that σ does, and map arbitrarily the remaining characters.

3.2 A 4-Approximation Algorithm for MPSM

Proposition 1. *There exists a 4-approximation algorithm for MPSM that runs in $O(n^{3/2})$ time.*

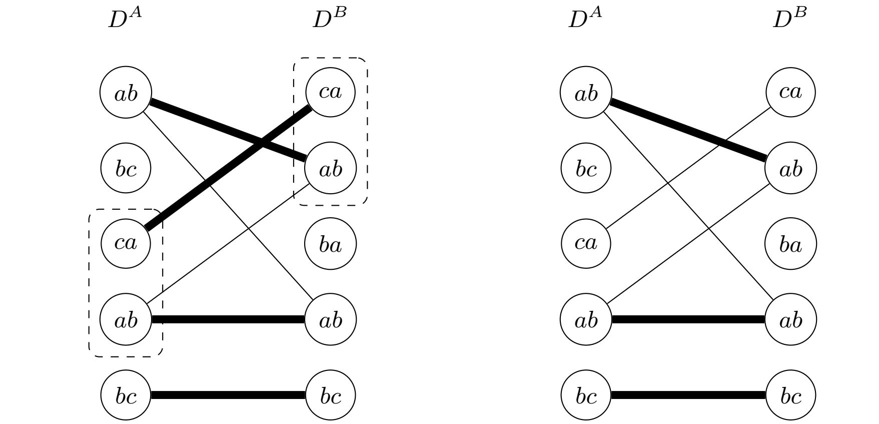
Proof. Consider the two strings $A = a_1 a_2 ... a_n$ and $B = b_1 b_2 ... b_n$ that one wishes to map while preserving a maximal number of duos, and let D^A and D^B denote their respective sets of duos. Also, denote by π^* an optimal mapping that preserves a maximum number $f(\pi^*)$ of duos.

Build a bipartite graph G in the following way: vertices on the left and the right represent duos of D^A and D^B, respectively. Whenever one duo on the right and one on the left are preservable (same two letters in the same order), we add an edge between the two corresponding vertices. Figure 3 provides an example of this construction.

At this point, notice that there exists a one-to-one correspondence between matchings in G and proper duo-mappings between D^A and D^B. In other words there exists a matching in G with $f(\pi^*)$ edges. Indeed, the set of duos preserved by any solution (and *a fortiori* by the optimal one) can be represented as a matching in G. Hence, denoting by M^* a maximum matching in G, it holds that :

$$f(\pi^*) \leqslant |M^*| \tag{1}$$

Unfortunately, a matching M^* in G does not immediately translate into a proper mapping that preserves $|M^*|$ duos. However, it does correspond to a proper duo-mapping that maps $|M^*|$ duos, which, as noticed earlier, might generate conflicts of Type 2 only.



(a) A proper duo-mapping with 2 conflicts (b) An unconflicting duo-mapping

Fig. 3. The graph G where $A = abcabc$ and $B = cababc$

In G, a conflict of Type 2 corresponds to two consecutive vertices on one side matched to two non-consecutive vertices on the other side. Hence, to generate an unconflicting duo-mapping σ using a matching M^*, it suffices to partition the matching M^* in 4 sub-matchings in the following way : Let $M(even, odd)$ denote the submatching of M^* containing all edges whose left endpoint have even indices, and right endpoint have odd indices; and define $M(odd, even)$, $M(even, even)$, and $M(odd, odd)$ in the same way. Denote by $\hat{M}$ the matching with biggest cardinality among these 4. Obviously, remembering that the four submatchings define a partition of M^*, it holds that $|\hat{M}| \geqslant |M^*|/4$. Considering that $\hat{M}$ does not contain any pair of edges with consecutive endpoints, the corresponding duo-mapping σ has no conflict. Following Remark 1, σ derives a proper mapping π on the characters such that:

$$f(\pi) \geqslant f(\sigma) = |\hat{M}| \geqslant \frac{|M^*|}{4} \overset{(1)}{\geqslant} \frac{f(\pi^*)}{4}$$

A 4-approximate solution can thus be computed by creating the graph G from strings A and B, computing an maximum matching M^* on it, partitioning M^* four ways by indices parity and return the biggest partition $\hat{M}$. Then map the matched duos following the edges $\hat{M}$, and map all the other characters arbitrarily. The complexity of the whole procedure is given by the complexity of computing an optimal matching in G, which is $O(n^{3/2})$. □

It is likely that the simple edge removal procedure that nullifies all conflicts of Type 2 can be replaced by a more involved heuristic method in order to solve efficiently real life problems.

3.3 An 8/5-Approximation for 2-MPSM

In the following, we make use of a reduction from MSPM to MAX INDEPENDENT SET (MIS) already pointed out in [13]. Given two strings A and B, consider the graph H built in the following way: H has a vertex v_{ij} for each preservable couple of duos $((D_i^A),(D_j^B))$, and H has an edge (v_{ij},v_{hl}) for each conflicting pair of preservable couple of duos $((D_i^A),(D_j^B))$ $((D_h^A),(D_l^B))$. It is easy to see that there is a 1 to 1 correspondence between independent sets in H and unconflicting duo-mappings between A and B.

Notice that, for a given k, a couple of duos $((D_i^A),(D_j^B))$ can belong to at most $6(k-1)$ conflicting pairs: on the one hand, there can be at most $2(k-1)$ conflicts of Type 1 (one for each other occurrence of the duo D_i^A in D^A and D^B), and on the other hand at most $4(k-1)$ conflicts of Type 2 (one for each possible conflicting occurrence of D_{j-1}^B or D_{j+1}^B in D^A, and one for each possible conflicting occurrence of D_{j-1}^A or D_{j+1}^A in D^B). This bound is tight.

Hence, for a given instance of k-MPSM, the corresponding instance of MIS is a graph with maximum degree $\Delta \leqslant 6(k-1)$. Using the approximation algorithm of [2] and [1] for independent set (which guarantees approximation ratio $(\Delta+3)/5$), this leads to obtaining approximation ratio arbitrarily close to $(6k-3)/5$ for k-MPSM, which already improves on the best known 2-approximation when $k=2$, and also on the 4-approximation of Proposition 1 when $k=3$.

We now prove the following result in order to further improve on the approximation:

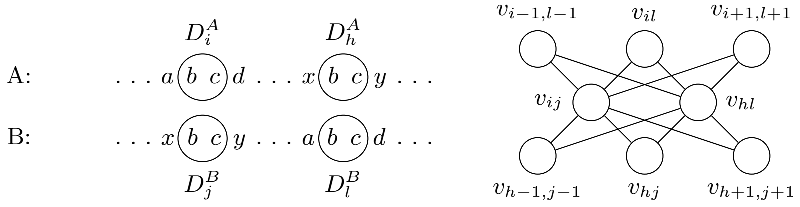
Lemma 1. *In a graph H corresponding to an instance of* 2*-MPSM, there exists an optimal solution for MIS that does not pick any vertex of degree* 6*.*

Proof. Consider a vertex v_{ij} of degree 6 in such a graph H. This vertex corresponds to a preservable couple of duos that conflicts with 6 other preservable couples. There exists only one possible configuration in the strings A and B that can create this situation, which is illustrated in Figure 4(a).

In return, this configuration always corresponds to the gadget illustrated in Figure 4(b), where vertices v_{ij}, v_{hj}, v_{il}, and v_{hl} have no connection with the rest of the graph.

Now, consider any maximal independent set S that picks some vertex v_{ij} of degree 6 in H. The existence of this degree-6 vertex induces that graph H contains the gadget of Figure 4(a). S is maximal, so it necessarily contains vertex v_{hl} as well. Let $S' = S \setminus (\{v_{ij}\},\{v_{hl}\}) \cup (\{v_{il}\},\{v_{hj}\})$. Reminding that v_{il} and v_{hj} have no neighbor outside of the gadget, it is clear that S' also defines an independent set.

Hence, in a maximal (and *a fortiori* optimal) independent set, any pair of degree-6 vertices (in such graphs, degree-6 vertices always appear in pair) can



(a) Strings A and B

(b) Gadget in H

Fig. 4. A degree 6-vertex in graph H

be replaced by a pair of degree 2 vertices, which concludes the proof of Lemma 1. □

Let H' be the subgraph of H induced by all vertices apart from vertices of degree 6. Lemma 1 tells us that an optimal independent set on H' has the same cardinality than an optimal independent set in H. However H' has maximum degree 5 and not 6, which yields a better approximation when using the algorithm described in [2] and [1]:

Proposition 2. *2-MPSM is approximable within ratio arbitrarily close to* $8/5$ *in polynomial time.*

Notice that the reduction from k-MPSM to $6(k-1)$-MIS also yields the following simple parameterized algorithm:

Corollary 1. *k-MPSM can be solved in $O^*((6(k-1)+1)^\psi)$, where ψ denotes the value of an optimal solution.*

Consider an optimal independent set S in H, if some vertex v of H has no neighbour in S then v necessarily belongs to S. Thus, in order to build an optimal solution S, one can go through the decision tree that, for each vertex v that has no neighbour in the current solution, consists of deciding which vertex among v and its set of neighbours will be included in the solution. Any solution S will take one of these $6(k-1)+1$ vertices. Each node of the decision tree has at most $6(k-1)+1$ branches, and the tree has obviously depth ψ, considering that one vertex is added to S at each level.

4 Some Results on 3-CONSTRAINED MAXIMUM INDUCED SUBGRAPH

In this section we consider the CONSTRAINED MAXIMUM INDUCED SUBGRAPH problem (CMIS) which is a generalization of MAX DUO-PRESERVATION STRING MAPPING (MPSM). In [6] the CMIS served as the main tool to analyze its special case, namely the MPSP problem. The problem is expressed as a natural linear program denoted by NLP, which is used to obtain a randomized k^2 approximation algorithm. In this section we provide a 6-approximation algorithm for the 3-CMIS which improves on the previous 9-approximation algorithm. We do this by introducing a *configuration-LP* denoted by CLP. Moreover we show that both NLP and CLP have an integrality gap of at least k which implies that it is unlikely to construct a better than k-approximation algorithm based on these linear programs.

We start with a formal definition of the problem.

Definition 2. CONSTRAINED MAXIMUM INDUCED SUBGRAPH *(CMIS):*

- ***Instance:*** *an m-partite graph $G(V,E)$ with parts: $G_1, \ldots, G_m$. Each part G_i has n_i^2 vertices organized in an $n_i \times n_i$ grid.*
- ***Solution:*** *a subset of vertices such that within each grid in each column and each row at most one vertex is chosen.*
- ***Objective:*** *maximizing the number of edges in the induced subgraph.*

In the constrained k-CMIS problem each grid consists of at most $k \times k$ vertices.

Let v_p^{ij} be the vertex placed in position (i,j) in the pth grid. Consider the linear program NLP as proposed in [6]. Let x_p^{ij} be the boolean variable which takes value 1 if the corresponding vertex v_p^{ij} is chosen, and 0 otherwise. Let $x_{p_{ij}q_{kl}}$ be the edge-corresponding boolean variable such that it takes the value 1 if both the endpoint vertices v_p^{ij} and v_q^{kl} are selected and 0 otherwise. The task is to choose a subset of vertices, such that within each block, in each column and each row at most one vertex is chosen. The objective is to maximize the number of edges in the induced subgraph. The LP formulation is the following:

$$\begin{array}{lll}
NLP: & & \\
Max & \displaystyle\sum_{\left(v_p^{ij} v_q^{kl}\right) \in E} x_{p_{ij}q_{kl}} & \\
s.t. & x_{p_{ij}q_{kl}} \leqslant x_p^{ij} & \text{for } i,j,k,l = [n_p], \quad p,q = [m], \\
 & \displaystyle\sum_{i=1}^{n_p} x_p^{ij} = 1 & \text{for } j = [n_p], \quad p = [m], \\
 & \displaystyle\sum_{j=1}^{n_p} x_p^{ij} = 1 & \text{for } i = [n_p], \quad p = [m], \\
 & 0 \leqslant x_{p_{ij}q_{kl}} \leqslant 1 & \text{for } i,j,k,l = [n_p], \quad p,q = [m], \\
 & 0 \leqslant x_p^{ij} \leqslant 1 & \text{for } i,j = [n_p], \quad p = [m].
\end{array} \qquad (2)$$

Note that when the size of each grid is constant, the CLP is of polynomial size. The first constraint ensures that the value of the edge-corresponding variable is not greater than the value of the vertex-corresponding variable of any of its endpoints. The second and the third constraints ensure that within each grid at most one vertex is taken in each column, each row, respectively.

Notice that within each grid there are $k!$ possible ways of taking a feasible subset of vertices. We call a *configuration*, a feasible subset of vertices for a given grid. Let us denote by $\mathcal{C}_p$ the set of all possible configurations for a grid p. Now, consider that we have boolean variable x_{C_p} for each possible configuration. The variable x_{C_p} takes value 1 if all the vertices contained in C_p are chosen and 0 otherwise. The induced linear program is called *Configuration-LP*, (CLP). The CLP formulation for the $CMIS$ problem is the following:

$$\begin{array}{lll} CLP(K): & & \\ Max & \displaystyle\sum_{\left(v_p^{ij} v_q^{kl}\right)\in E} x_{p_{ij}q_{kl}} & \\ s.t. & x_{p_{ij}q_{kl}} \leqslant x_p^{ij} & \text{for } i,j,k,l = [n_p], \quad p,q = [m], \\ & x_p^{ij} = \displaystyle\sum_{v_p^{ij}\in C_p \in \mathcal{C}_p} x_{C_p} & \text{for } i,j = [n_p], \quad p, = [m], \\ & \displaystyle\sum_{C_p\in\mathcal{C}_p} x_{C_p} = 1 & \text{for } p = [m], \\ & 0 \leqslant x_{p_{ij}q_{kl}} \leqslant 1 & \text{for } i,j,k,l = [n_p], \quad p,q = [m], \\ & 0 \leqslant x_{C_p} \leqslant 1 & \text{for } C_p \in \mathcal{C}_p, \quad p = [m], \end{array} \tag{3}$$

The first constraint is the same as in NLP. The second one ensures that the value of the vertex-corresponding variable is equal to the summation of the values of the configuration-corresponding variables containing considered vertex. The third constraint ensures that within each grid exactly one configuration can be taken. Notice that the vertex variables are redundant and serve just as an additional description. In particular the first and the second constraints could be merged into one constraint without vertex variables.

One can easily see that the CLP is at least as strong as the NLP formulation: a feasible solution to CLP always translates to a feasible solution to NLP.

Proposition 3. *There exists a randomized 6-approximation algorithm for the* 3-CONSTRAINED MAXIMUM INDUCED SUBGRAPH *problem.*

Proof. Consider a randomized algorithm that, in each grid G_p, takes the vertices from configuration C with a probability $\frac{\sqrt{x_C}}{\sum_{C_p\in\mathcal{C}_p}\sqrt{x_{C_p}}}$.

Consider any vertex, w.l.o.g. $v_p^{1,1}$. Each vertex is contained in two configurations, w.l.o.g. let $v_p^{1,1}$ be contained in C_p^1 and C_p^2. The probability that $v_p^{1,1}$ is chosen is:

$$\Pr\left(v_p^{1,1}\text{is taken}\right) = \frac{\sqrt{x_{C_p^1}} + \sqrt{x_{C_p^2}}}{\sum_{C_p\in\mathcal{C}_p}\sqrt{x_{C_p}}}$$

Optimizing the expression $\sqrt{x_{C_p^1}} + \sqrt{x_{C_p^2}}$ under the condition $x_{C_p^1} + x_{C_p^2} = x_p^{1,1}$, we have that the minimum is when either $x_{C_p^1} = 0$ or $x_{C_p^2} = 0$ which implies $\sqrt{x_{C_p^1}} + \sqrt{x_{C_p^2}} = \sqrt{x_p^{1,1}}$. Thus:

$$\Pr\left(v_p^{1,1}\text{is taken}\right) \geqslant \frac{\sqrt{x_p^{1,1}}}{\sum_{C_p\in\mathcal{C}_p}\sqrt{x_{C_p}}}$$

Using a standard arithmetic inequality we can get that:

$$\frac{\sum_{C_p \in \mathcal{C}_p} \sqrt{x_{C_p}}}{6} \leqslant \sqrt{\frac{\sum_{C_p \in \mathcal{C}_p} x_{C_p}}{6}} = \sqrt{\frac{1}{6}}$$

which implies that:

$$\Pr\left(v_p^{1,1}\text{is taken}\right) \geqslant \frac{\sqrt{x_p^{1,1}}}{\sqrt{6}}$$

Now let us consider any edge and the corresponding variable, $x_{p_{ij}q_{kl}}$. The probability that the edge is taken can be lower bounded by:

$$\Pr\left(x_{p_{ij}q_{kl}}\text{is taken}\right) = \Pr\left(v_p^{ij}\text{is taken}\right) \cdot \Pr\left(v_q^{kl}\text{is taken}\right) \geqslant \frac{\sqrt{x_p^{ij}}}{\sqrt{6}} \cdot \frac{\sqrt{x_q^{kl}}}{\sqrt{6}} \geqslant$$

$$\frac{1}{6}\min\{x_p^{ij}, x_q^{kl}\} \geqslant \frac{1}{6} x_{p_{ij}q_{kl}}$$

Since our algorithm takes in expectation every edge with probability $\frac{1}{6}$ of the fractional value assigned to the corresponding edge-variable by the CLP it is a randomized 6-approximation algorithm. □

4.1 Integrality Gap of *NLP* and *CLP*

We now show that the linear relaxation NLP has an integrality gap of at least k. Consider the following instance of k-CMIS. Let the input graph $G(V, E)$ consists of two grids, G_1, G_2. Both grids consist of k^2 vertices. Every vertex from one grid is connected to all the vertices in the second grid and vice versa. Thus the number of edges is equal to k^4. By putting all the LP variables to $\frac{1}{k}$ one can easily notice that this solution is feasible and the objective value for this solution is k^3. On the other hand any feasible integral solution for this instance must return at most k vertices from each grid, each of which is connected to at most k vertices from the other grid. Thus the integral optimum is at most k^2. This produces the intergality gap of k. Moreover by putting the configuration-corresponding variables in $CLP(K)$ to $\frac{1}{k!}$ we can construct a feasible solution to $CLP(K)$ with the same integrality gap of k.

References

1. Berman, P., Fujito, T.: On Approximation Properties of the Independent Set Problem for Low Degree Graphs. Theory of Computing Systems 32(2), 115–132 (1999)
2. Berman, P., Fürer, M.: Approximating Maximum Independent Set in Bounded Degree Graphs. In: Sleator, D.D. (ed.) SODA, pp. 365–371. ACM/SIAM (1994)
3. Berman, P., Karpinski, M.: On Some Tighter Inapproximability Results (Extended Abstract). In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 200–209. Springer, Heidelberg (1999)

4. Bulteau, L., Fertin, G., Komusiewicz, C., Rusu, I.: A Fixed-Parameter Algorithm for Minimum Common String Partition with Few Duplications. In: Darling, A., Stoye, J. (eds.) WABI 2013. LNCS, vol. 8126, pp. 244–258. Springer, Heidelberg (2013)
5. Chen, J., Kanj, I.A., Jia, W.: Vertex Cover: Further Observations and Further Improvements. In: Widmayer, P., Neyer, G., Eidenbenz, S. (eds.) WG 1999. LNCS, vol. 1665, pp. 313–324. Springer, Heidelberg (1999)
6. Chen, W., Chen, Z., Samatova, N.F., Peng, L., Wang, J., Tang, M.: Solving the maximum duo-preservation string mapping problem with linear programming. Theoretical Computer Science 530, 1–11 (2014)
7. Chen, X., Zheng, J., Fu, Z., Nan, P., Zhong, Y., Lonardi, S., Jiang, T.: Assignment of Orthologous Genes via Genome Rearrangement. Transactions on Computational Biology and Bioinformatics 2(4), 302–315 (2005)
8. Chrobak, M., Kolman, P., Sgall, J.: The Greedy Algorithm for the Minimum Common String Partition Problem. In: Jansen, K., Khanna, S., Rolim, J.D.P., Ron, D. (eds.) RANDOM 2004 and APPROX 2004. LNCS, vol. 3122, pp. 84–95. Springer, Heidelberg (2004)
9. Cormode, G., Muthukrishnan, S.: The string edit distance matching problem with moves. ACM Transactions on Algorithms 3(1) (2007)
10. Damaschke, P.: Minimum Common String Partition Parameterized. In: Crandall, K.A., Lagergren, J. (eds.) WABI 2008. LNCS (LNBI), vol. 5251, pp. 87–98. Springer, Heidelberg (2008)
11. Downey, R.G., Fellows, M.R.: Parameterized Complexity, p. 530. Springer (1999)
12. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Co., San Francisco (1979)
13. Goldstein, A., Kolman, P., Zheng, J.: Minimum Common String Partition Problem: Hardness and Approximations. In: Fleischer, R., Trippen, G. (eds.) ISAAC 2004. LNCS, vol. 3341, pp. 484–495. Springer, Heidelberg (2004)
14. Jiang, H., Zhu, B., Zhu, D., Zhu, H.: Minimum common string partition revisited. Journal of Combinatorial Optimization 23(4), 519–527 (2012)
15. Kolman, P., Walen, T.: Approximating reversal distance for strings with bounded number of duplicates. Discrete Applied Mathematics 155(3), 327–336 (2007)
16. Kolman, P., Walen, T.: Reversal Distance for Strings with Duplicates: Linear Time Approximation using Hitting Set. Electronic Journal of Combinatorics 14(1) (2007)
17. Bulteau, L., Komusiewicz, C.: Minimum common string partition parameterized by partition size is fixed-parameter tractable. In: SODA, pp. 102–121 (2014)
18. Lund, C., Yannakakis, M.: The Approximation of Maximum Subgraph Problems. In: Lingas, A., Karlsson, R.G., Carlsson, S. (eds.) ICALP 1993. LNCS, vol. 700, pp. 40–51. Springer, Heidelberg (1993)
19. Swenson, K.M., Marron, M., Earnest-DeYoung, J.V., Moret, B.M.E.: Approximating the true evolutionary distance between two genomes. ACM Journal of Experimental Algorithmics 12 (2008)