Theoretical Computer Science 542 (2014) 98–107

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

# Quick greedy computation for minimum common string partition

Isaac Goldstein, Moshe Lewenstein *,[1]

*Department of Computer Science, Bar-Ilan University, Ramat Gan 52900, Israel*

ARTICLE INFO

*Article history:*
Received 21 November 2013
Received in revised form 1 May 2014
Accepted 10 May 2014
Communicated by M. Crochemore

*Keywords:*
Strings
Approximation algorithm
Pattern matching

ABSTRACT

In the *minimum common string partition* problem one is given two strings $S$ and $T$ with the same character statistics and one seeks for the smallest partition of $S$ into substrings so that $T$ can also be partitioned into the same substring multiset. The problem is fundamental in several variants of edit distance with block operations, e.g. signed reversal distance with duplicates and edit distance with moves.

The minimum common string partition problem is known to be NP-complete and the best approximation algorithm known has an approximation factor of $O(\log n \log^* n)$. Since the minimum common string partition problem is of utmost practical importance one seeks a heuristic that will (1) usually have a low approximation factor and (2) will run fast.

A simple greedy algorithm is known, which iteratively choose non-overlapping longest common substrings of the input strings. This algorithm has been well-studied from an approximation point of view and it has been shown to have a bad worst case approximation factor. However, all the bad approximation factors presented so far stem from complicated recursive construction. In practice the greedy algorithm seems to have small approximation factors. However, the best current implementation of greedy runs in quadratic time.

We propose a novel method to implement greedy in linear time.



## 1. Introduction

The classical edit distance is the number of edit operations, insertions, deletions, character exchanges and (sometimes) swaps, needed to transform one string into another. All the edit operations work on single characters, besides swap which operates on two. Motivation for block operations has stimulated a large collection of new problems, some of which have a completely different flavor than the original edit distance problem.

Sankoff and Kruskal [21] and Tichy [23] were the first to consider block operations together with simple character operations. Lopresti and Tomkins [19] suggested several distance measures for block operations in the same spirit as they were defined for the case of single characters. The subject of block edit distance gained an increasing interest in the last decade especially due to its important impact on the computational biology field. In the research of the homology of structures there is an interest in finding organisms' characteristics that are derived from a common ancestor. The task of identifying the orthologs, especially direct descendants of ancestral genes in current species is a fundamental problem in computational biology. A promising approach to solving this problem is to take into account not only local mutations but

* Corresponding author.
*E-mail addresses:* goldshi@cs.biu.ac.il (I. Goldstein), moshe@cs.biu.ac.il (M. Lewenstein).

[1] This research was supported by BSF grant 2010437 and GIF grant 1147/2011.

http://dx.doi.org/10.1016/j.tcs.2014.05.006
0304-3975/

also global genome rearrangements events. This approach is strongly connected to the area of edit distance where block operations are allowed, since those genes can be represented as sequences of characters with block operations applied to them [4].

### 1.1. Sorting by reversals and MCSP

Chen et al. [4] studied the problem of signed reversal distance with duplicates (SRDD) which is a slight extension of the well-known sorting by reversals (SBR) problem. They showed that this problem is NP-hard even for $k = 2$, where $k$ is the maximum number of occurrences of any character in the input strings (it is worth noting that in the unsigned case the problem is NP-hard even if $k = 1$. However, for the signed case there is a polynomial algorithm for solving SBR for $k = 1$, see [14]). Moreover, they pointed out that this problem is closely related to the minimum common string partition problem (MCSP). In the MCSP we are given two strings $S$ and $T$ and we need to find a partition of each of these strings to substrings, so that the set of substrings of $S$ and $T$ are the same. We define this more fully in Section 2.

### 1.2. Edit distance with moves and MCSP

The problem of edit distance with moves was studied by Cormode and Muthukrishnan [8]. In this variant of the classical edit distance, moving a substring is allowed in addition to deleting and inserting single characters. They showed the problem to be NP-complete and suggested an approximation algorithm with an approximation factor of $O(\log n \log^* n)$ and running time $O(n \log^* n)$. This is still the best approximation algorithm given for the problem. Lately there has been a renewed interest in approximating NP-complete problems for substring, subsequence and supersequence problems [1,7,12,13].

Shapira and Storer [22] observed that the problem, while recursive, can be transformed into a non-recursive version with a constant-factor cost in the approximation. Moreover, they showed that the problem can be transformed into a version in which moves only are allowed. It has been observed that this is in essence a reduction to the MCSP problem, although not specifically noted so in the paper.

Using Shapira and Storer's observations one can use Cormode and Muthukrishnan [8] approximation algorithm for MCSP as well. This is currently the best known approximation.

### 1.3. Restricted versions of MCSP

Goldstein et al. [11] proved that even the simple case of 2-MCSP (MCSP in which each character occurs at most twice) is NP-hard. Moreover, they showed that it is APX-hard. They also gave an 1.1037-approximation algorithm for this problem which improved the 1.5 ratio showed by Chen et al. [4]. For 3-MCSP a 4-approximation algorithm was given [11]. Kolman [17] proposed an $O(k^2)$-approximation algorithm for $k$-MCSP with $O(nk)$ running time for general $k$, where each symbol can occur at most $k$ times. Kolman and Walen [18] improved this result and gave a $O(k)$-approximation algorithm with $O(n)$ running time.

Due to the close relation between the MCSP problem and the SBR problem (as shown by Chen et al. [4]) the approximation factor for these restricted versions of MCSP also applies to parallel restricted versions of SBR with just a constant factor penalty. Instead of restricting the problem by the number of each symbol occurrences, the MCSP problem and the related problem of SBR could be also restricted by the alphabet size. Unsigned SBR for unary alphabet could be solved trivially. However, Christie and Irving [5] showed that for binary alphabet unsigned SBR becomes NP-hard. Similar NP-hardness results for MCSP incorporating alphabet sizes greater than 1 were given by Jiang et al. [15].

### 1.4. MCSP parameterized

Recently there have been several attempts to parameterize the problem. Damaschke [9] showed that the problem of MCSP is fixed-parameter tractable (FPT) on both $k$, the size of the optimal solution to the MCSP problem of the input strings, and $r$, the repetition number of a string $x$, which is defined as the maximum $i$ so that $x = uv^i w$ holds for some strings $u$, $v$, $w$, with nonempty $v$. Jiang et al. [15] presents an FPT algorithm for the $d$-MCSP problem (MCSP where each symbol appears at most $d$ times) that runs in $O((d!)k)$ time. They also investigated the case in which the partition is $x$-balanced, which requires the length of each block not to be more than $x$ away of the average length. They devised a FPT algorithm with parameters $k$ and $x$ which runs in $O((2x)kk!n)$ time for this case. Following their work, Bulteau et al. [2] gave an FTP algorithm that run in $O(d^{2k}kn)$ time. Their algorithm also applies to a generalized version of the MCSP problem. Finally, Bulteau and Komusiewicz [3] showed a FTP algorithm parametrized only by $k$.

### 1.5. MCSP and the GREEDY algorithm

A simple greedy algorithm, to be denoted by GREEDY, was suggested by Shapira and Storer [22]. They showed that for large classes of inputs the algorithms have a logarithmic approximation factor.

Chrobak et al. [6] demonstrated that in the general case the approximation ratio of GREEDY for the MCSP is not better than $\Omega(n^{0.43})$. In addition, for the special case of 4-MCSP it was found that it is at least $\Omega(\log n)$. Nevertheless, they showed it has an $O(n^{0.69})$ approximation factor for the general case, and a 3-approximation factor for 2-MCSP. The lower bound on GREEDY's performance for the general case was later increased to $\Omega(n^{0.46})$ by Kaplan and Shafrir [16].