On the Minimum Common Integer Partition Problem

XIN CHEN

Nanyang Technological University

LAN LIU

Google Inc.

ZHENG LIU

City of Hope, National Medical Center

and

TAO JIANG

University of California at Riverside

We introduce a new combinatorial optimization problem in this paper, called the *Minimum Common Integer Partition* (MCIP) problem, which was inspired by computational biology applications including ortholog assignment and DNA fingerprint assembly. A *partition* of a positive integer n is a multiset of positive integers that add up to exactly n, and an *integer partition* of a multiset S of integers is defined as the multiset union of partitions of integers in S. Given a sequence of multisets $S_1, S_2, \cdots, S_k$ of integers, where $k \geq 2$, we say that a multiset is a *common integer partition* if it is an integer partition of every multiset S_i, $1 \leq i \leq k$. The MCIP problem is thus defined as to find a common integer partition of $S_1, S_2, \cdots, S_k$ with the minimum cardinality, denoted as $\mathrm{MCIP}(S_1, S_2, \cdots, S_k)$. It is easy to see that the MCIP problem is NP-hard since it generalizes the well-known Subset Sum problem. We can in fact show that it is APX-hard. We will also present a $\frac{5}{4}$-approximation algorithm for the MCIP problem when $k = 2$, and a $\frac{3k(k-1)}{3k-2}$-approximation algorithm for $k \geq 3$.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms: Algorithms

Additional Key Words and Phrases: Subset sum, integer partition, NP-hard, approximation algorithm, computational biology, combinatorial optimization

---

1. INTRODUCTION

Computational molecular biology has emerged as one of the most exciting inter-disciplinary fields in the past two decades, in part because various biological applications have spawned a large number of interesting combinatorial problems such as multiple sequence alignment [Gusfield 1997], sorting by reversals [Sankoff 1989], and recently the minimum common partition problem [Chen et al. 2005a]. These problems have attracted considerable attention from computer scientists who took the challenge to design efficient and effective algorithms for solving them [Altschul and Lipman 1989; Hannenhalli and Pevzner 1995; Goldstein et al. 2004]. In this paper, we introduce a new combinatorial optimization problem, called the *Minimum Common Integer Partition* problem (MCIP), which was inspired by our recent work on ortholog assignment and DNA fingerprint assembly.

By a *partition* of a positive integer n we mean a multiset [1] $\{n_1, n_2, \cdots, n_r\}$ of positive integers that add up to exactly n, *i.e.* $\sum_{i=1}^{r} n_i = n$, where n_i is called a *part* of the partition of n [Andrews 1976; Andrews and Eriksson 2004]. We will use $P(n)$ to denote a specific partition of n. Given a multiset $S = \{x_1, x_2, \cdots, x_m\}$ of integers with a partition $P(x_i)$ for each integer x_i, $1 \leq i \leq m$, we can define an *integer partition* of S as the multiset union of these partitions, that is $\biguplus_{i=1}^{m} P(x_i)$. By definition, S is an integer partition of itself. A multiset is said to be a *common integer partition* of a sequence of multisets $S_1, S_2, \ldots, S_k (k \geq 2)$ if it is an integer partition of every multiset S_i, $1 \leq i \leq k$. The minimum common integer partition problem is thus defined as follows: given a sequence of multisets $S_1, S_2, \cdots, S_k$ of integers, find a common integer partition of them with the minimum cardinality. We denote the minimum common integer partition by $\mathrm{MCIP}(S_1, S_2, \cdots, S_k)$ (or simply MCIP when the input multisets are clear from the context). Note that, now MCIP denotes both the MCIP problem and also its solution on a particular instance, but this overloading of the notation is a common pratice and should not cause any confusion given the context. For simplicity, we also denote by $\mathrm{MCIP}(S_1, S_2, \cdots, S_k)$ (or simply k-MCIP) the restricted version of the MCIP problem when the number of input multisets is fixed to be k throughout the paper.

For example, the integer 3 has only three partitions, *i.e.*, $\{3\}, \{2, 1\}$, and $\{1, 1, 1\}$, while the integer 10 has 190569292 partitions [Andrews 1976]. We can see that the number of partitions increases quite rapidly with the integer n. For multiset $S = \{3, 3, 4\}$, $\{2, 2, 3, 3\}$ is an integer partition of S and $\{1, 1, 2, 2, 4\}$ is another one. For a pair of multisets $S = \{3, 3, 4\}$ and $T = \{2, 2, 6\}$, both $\{2, 2, 3, 3\}$ and $\{1, 1, 2, 2, 4\}$ are common integer partitions of S and T, while the first one gives the minimum cardinality, *i.e.*, $\mathrm{MCIP}(S, T) = \{2, 2, 3, 3\}$. Note that the minimum common integer partition is not necessarily unique. So, the notation $\mathrm{MCIP}(S_1, S_2, \cdots, S_k)$ is not really a function, strictly speaking. But we will use it as a function throughout the paper for simplicity.

The necessary and sufficient condition for a sequence of multisets $S_1, S_2, \ldots, S_k$ to have a common integer partition is that they have the same summation over their integer elements. Multisets with this property are called *related*. Verifying

---

[1] Recall that a multiset is a set-like object in which order is ignored, but multiplicity is explicitly significant.

whether a sequence of multisets of integers are related can be done easily in linear time, and thus for the rest of the paper we will assume, without loss of generality, that the input multisets are all related.

Clearly, the MCIP problem is NP-hard since it generalizes the well-known *Subset Sum* problem [Cormen et al.]. The reduction from the Subset Sum problem to the MCIP problem is based on the following observation: given a set $S = \{x_1, ..., x_n\}$ of n positive integers and a positive integer x, there exists a subset of S with sum x, if and only if the MCIP solution for multisets S and $T = \{x, (\sum_{x_i \in S} x_i - x)\}$ has the cardinality n. We also present a $\frac{5}{4}$-approximation algorithm for the 2-MCIP problem using a heuristic for the *Maximum Set Packing* problem, and a $\frac{3k(k-1)}{3k-2}$-approximation algorithm for the general k-MCIP problem, where $k \geq 3$. In this paper, we show that the MCIP problem is APX-hard and hence has no polynomial-time approximation algorithm (PTAS) unless P = NP.

1.1   Biological Background

Although the MCIP problem is quite a natural extension of the Subset Sum problem, its formulation was mainly motivated by our recent work on ortholog assignment and DNA fingerprint assembly in computational molecular biology. The following gives a brief account of the background. Since it contains discussions that involve the knowledge of some biological experiments, the reader who is not interested in the biological relevance may feel free to skip some (or all) of the paragraphs in this subsection.

Ortholog assignment. Orthologous genes are typically the evolutionary and functional counterparts in different species, and therefore the prediction (or assignment) of orthologs is a common task in computational biology. While it is usually done using sequence homology search [Remm et al. 2001], we have recently proposed an alternative and promising approach to assign orthologs via genome rearrangement [Chen et al. 2005b; 2005a]. This new approach has inspired us to formulate several interesting combinatorial optimization problems, *e.g.*, Signed Reversal Distance with Duplicates (SRDD), Minimum Common String Partition (MCSP), and Maximum Cycle Decomposition (MCD), which have attracted increasing attention from the algorithms community [Chrobak et al. 2004; Goldstein et al. 2004; Fu 2004; Kolman 2005]. In particular, the MCSP problem, which is the most related to MCIP, is defined as follows: given two input strings, partition them into the same collection of substrings so that the number of resultant substrings is minimized. For example, the MCSP for $\{aaabbbccc, bbbaaaccc\}$ is $\{aaa, bbb, ccc\}$. The restricted version of MCSP where the number of symbols that occur in an input string multiple times (called duplicated symbols; the other symbols are called singletons) is no more than l in each input string, is denoted by MCSP-l. It is known that the MCSP-l problem is NP-hard [Chen 2005], when $l \geq 1$. In other words, even when there is only one symbol with multiple copies in each input string, we still cannot find the MCSP in polynomial time unless P=NP.

It is easy to transform an instance of MCSP-1 into an instance of 2-MCIP where each integer represents the size of a block consisting of only the duplicated symbol so that an optimal solution to the 2-MCIP problem would in most cases give an optimal solution to the MCSP-1 problem with the same cardinality [Chen 2005].

Therefore, we hope that the study of MCIP will help the design of good approximation algorithms for MCSP-1 and MCSP in general.

DNA fingerprint assembly. In the ongoing *Oligonucleotide Fingerprinting Ribosomal Genes* (OFRG) project [Valinsky et al. 2004], we collaborate with microbiologists and statisticians to provide a high-throughput method for identifying different microbial organisms. Briefly, the microbiologists build an rDNA clone library after DNA extraction and *Polymerase Chain Reaction* (PCR) amplification. The rDNA clones are assigned fingerprints (binary strings where 0 indicate non-binding between a clone and a probe, and 1 otherwise) through a series of hybridization experiments, each using a single 10-nucleotide DNA probe. These 10-nucleotide DNA probes comprise a probe set and the size of the probe set determines the length of a fingerprint. Then, clones are identified by clustering their fingerprints with those of known sequences. By mapping sequence data to hybridization patterns, clones can be identified (or at least differentiated). Compared with direct sequencing, the method saves significant cost without sacrificing too much discriminating ability.

Although OFRG is a cost-effective approach, we are trying to scale it up in order to process a large number of samples from applications such as identifying microorganisms involved in the development of the mucosal and systemic immune system. One possible way of enhancing OFRG is inspired by new (but proven) technologies such as microbead clone libraries and multiplex flow cytometry. By producing clone libraries on microbeads, we are able to simultaneously hybridize a set of probes to thousands of clones in seconds, which is a significant improvement over the current array platform. However, we will still need multiple hybridizations, each using a different probe (sub)set, as the size of the desired probe set in OFRG exceeds the maximum discriminating size of the cytometry technology. Thus we obtain a *partial fingerprint* from each run of hybridization because only a subset of the probes are used in each hybridization.

The *DNA fingerprint assembly problem* aims at inferring a *complete fingerprint* (with respect to the overall probe set) for each clone from partial fingerprints by minimizing the total number of distinct complete fingerprints. We assume that all the probe subsets share a small number of common probes which are called the *linking probes*. That is, these linking probes will be used for each run of hybridization. A complete fingerprint can thus be obtained from partial fingerprints that share the same bits on the linking probes. More specifically, after each run of the hybridization, we assign a *weight* to each distinct partial fingerprint as the number of clones that produced this partial fingerprint in the hybridization. Then we divide all partial fingerprints into groups based on their bits on the positions of linking probes. The partial fingerprints in a group are compatible with each other and may correspond to the same complete fingerprint. For each group, the fingerprint assembly problem can be viewed as $\mathrm{MCIP}(S_1, S_2, \cdots, S_k)$, with k being the number of the probe subsets (*i.e.* the number of hybridizations) and S_i containing the weight of each partial fingerprint in this particular group from the ith hybridization. Hence, complete fingerprints for each group can be obtained by combining their respective partial fingerprints via the minimum common integer partition of the weights. Such a solution would represent the minimum number of *distinct* complete fingerprints (or clones) that have produced the group of partial

fingerprints.

2. SOME BASIC FACTS

Throughout the paper, we assume that the multisets given as input to MCIP are related as mentioned before. We denote the size of the minimum common integer partition by $|MCIP(S_1, S_2, \cdots, S_k)|$ (or simply $|k\text{-}MCIP|$ if the input multisets are clear from the context). Because every integer in any input multiset will be partitioned into one or more integers in the minimum common integer partition, the following lemma gives a trivial, but useful lower bound.

LEMMA 2.1. $|MCIP(S_1, S_2, \cdots, S_k)| \geq max(|S_1|, |S_2|, \cdots, |S_k|)$, where $|\cdot|$ is the size of a multiset.

PROOF. The proof is obvious and thus omitted. □

In the case of 2-MCIP, we use $\langle S, T \rangle$ to denote the two input multisets, where $S = \{x_1, x_2, \cdots, x_m\}$ and $T = \{y_1, y_2, \cdots, y_n\}$ such that $\sum_{i=1}^{m} x_i = \sum_{i=1}^{n} y_i$. A greedy algorithm that constructs a common integer partition of $\langle S, T \rangle$ is to iteratively add the smaller one of two integers randomly selected from the two input multisets. More precisely, the algorithm can be described in pseudo-code as in Figure 1, and runs in time linear in n. The following lemma gives an upper bound for 2-MCIP, which is very useful in the subsequent discussion.

LEMMA 2.2. $|MCIP(S, T)| \leq |S| + |T| - 1$.

PROOF. After each iteration of the algorithm 2-APPROX-MCIP(S,T), the total size of S and T shall decrease by one or two while the multiset CIP expands by one integer. In the last iteration, the two integers remaining in S and T must be equal, and thus the total size of S and T shall decrease by exactly two. Therefore, the common integer partition returned from the algorithm contains no more than $|S| + |T| - 1$ integers. □

As its name suggests, the algorithm 2-APPROX-MCIP(S,T) is a 2-approximation algorithm for the problem of 2-MCIP.

LEMMA 2.3. *The algorithm* 2-APPROX-MCIP*(S,T) achieves an approximation ratio of 2.*

PROOF. The result is trivially implied by Lemma 2.1 and Lemma 2.2. □

Given a common integer partition $CIP(S, T)$ of $\langle S, T \rangle$, we say that x_i is *mapped* to y_j if there exists an element in $CIP(S, T)$ such that it is a part of the partition of x_i as specified in $CIP(S, T)$ as well as a part of the partition of y_j. Notice that an integer in S (or T) can be mapped to two or more integers in T (or S). Two integers a_1 and a_h in $\langle S, T \rangle$, *i.e.*, $a_1 \in S \uplus T$ and $a_h \in S \uplus T$, are said to be *connected* if there exist a sequence of integers $a_2, \cdots, a_{h-1}$ in $\langle S, T \rangle$ such that a_i is mapped to a_{i+1}, for each $i \in [1, h-1]$. Thus, all the integers that are connected to each other in S and T will constitute a *connected component* (or simply *component*) of $\langle S, T \rangle$. We say that these connected components are *induced* by the given common integer partition $CIP(S, T)$.

```
Algorithm 2-Approx-MCIP(S, T)

input Related multisets S = {x_1, ⋯, x_m}
         and T = {y_1, ⋯, y_n}
output A common integer partition CIP of S and T
begin
    CIP := ∅;
    while S ≠ ∅ do
        arbitrarily pick x_i ∈ S and y_j ∈ T;
        S := S \ {x_i};              // remove x_i from S;
        T := T \ {y_j};              // remove y_j from T;
        if x_i < y_j then
            CIP := CIP ⊎ {x_i};      // add x_i to CIP;
            y_j := y_j − x_i;
            T := T ⊎ {y_j};          // add y_j to T;
        else if x_i > y_j then
            CIP := CIP ⊎ {y_j};      // add y_j to CIP;
            x_i := x_i − y_j;
            S := S ⊎ {x_i};          // add x_i to S;
        else if x_i = y_j then
            CIP := CIP ⊎ {x_i};      // add x_i to CIP;
    return CIP;
end.
```

Fig. 1.   A 2-approximation algorithm for 2-MCIP.

LEMMA 2.4. *Suppose that $CIP(S,T)$ denotes a common integer partition of S and T. Then*

(1) *every connected component $\langle S_1, T_1 \rangle$ induced by $CIP(S,T)$ is a pair of related multisets;*

(2) *for every connected component $\langle S_1, T_1 \rangle$, all the integers in $CIP(S,T)$ that are partition parts of the integers in S_1 or T_1 constitute a common integer partition $CIP(S_1, T_1)$ of S_1 and T_1 such that $|CIP(S_1, T_1)| \geq |S_1| + |T_1| - 1$.*

PROOF. (1) Based on the common integer partition $CIP(S,T)$, each part of the partition of an integer x_i in S_1 corresponds to a distinct part of the partition of exactly one integer y_j in T in a one-to-one fashion. In this case, x_i and y_j are mapped to each other, and by the definition of connected components, y_j will be included in T_1, implying that $\sum_{x \in S_1} x \leq \sum_{y \in T_1} y$. Similarly, we have $\sum_{x \in S_1} x \geq \sum_{y \in T_1} y$. Therefore, S_1 and T_1 are two related multisets.

(2) Since the multiset under consideration (*i.e.* $CIP(S_1, T_1)$) consists of all integers from $CIP(S,T)$ that are partition parts of integers in S_1 or T_1 and nothing else, it is clearly a common integer partition of S_1 and T_1.

To see that $|CIP(S_1, T_1)| \geq |S_1| + |T_1| - 1$, we construct an undirected graph based on the integers in S_1, T_1 and $CIP(S_1, T_1)$: for each integer in S_1 (or T_1), a vertex is created and, for each integer in $CIP(S_1, T_1)$ which is a part of the partition of x_i as well as a part of the partition of y_j, an edge is created between the vertices for x_i and y_j. We denote by $|V|$ the number of vertices in the graph and by $|E|$ the number of edges. Observe that there is a one-to-one correspondence between the

vertices and the integers in $S_1 \uplus T_1$, and hence $|V| = |S_1| + |T_1|$. Further observe that there is a one-to-one correspondence between the edges and the integers in $CIP(S_1, T_1)$, and thus $|E| = |CIP(S_1, T_1)|$. We can see that the constructed graph is connected and may have multiple edges between a pair of vertices, which means that $|E| \geq |V| - 1$. Thus $|CIP(S_1, T_1)| \geq |S_1| + |T_1| - 1$ holds. $\square$

2.1   The Maximum Related Multiset Partition

In this subsection, we define a new combinatorial optimization problem, *maximum related multiset partition (MRMP)*, to assist solving the MCIP problem.

S_1 and T_1 are said to be a pair of *related submultisets* of two related multisets S and T if S_1 is a (nonempty) submultiset of S, T_1 is a (nonempty) submultiset of T, and they are related. We write $\langle S_1, T_1 \rangle \subseteq \langle S, T \rangle$ to denote the related submultisets. Obviously, $\langle S, T \rangle \subseteq \langle S, T \rangle$. Furthermore, S and T are said to be *basic* if they have one and only one pair of related submultisets, namely $\langle S, T \rangle$. For example, consider $S = \{3, 3, 4\}$ and $T = \{2, 2, 6\}$. They have three pairs of related submultisets: $\langle \{3, 3\}, \{6\} \rangle$, $\langle \{4\}, \{2, 2\} \rangle$, and $\langle S, T \rangle$. Therefore, S and T are not a pair of basic related multisets. An example of two basic related multisets is $\langle \{1, 4\}, \{2, 3\} \rangle$.

A *multiset partition* (or simply *partition*) of a multiset S is a sequence of disjoint submultisets $S_1, S_2, \cdots, S_l$ of S whose union is S, *i.e.* $S = \uplus_{i=1}^{l} S_i$. By definition, S is a multiset partition of itself. It is important to remember that multiset partition and the integer partition are two different concepts in this paper. Given two multisets S and T of integers, a sequence of multiset pairs $\langle S_1, T_1 \rangle, \langle S_2, T_2 \rangle, \cdots, \langle S_l, T_l \rangle$ is called a *related multiset partition* if $\{S_1, S_2, \cdots, S_l\}$ is a multiset partition of S, $\{T_1, T_2, \cdots, T_l\}$ is a multiset partition of T, and, moreover, for each $i \in [1, l]$, S_i and T_i are a pair of related multisets. The maximum related multiset partition problem is then defined as to find a related multiset partition of two given multisets S and T, maximizing the number of related multiset pairs in the partition. We denote by $MRMP(S, T)$ (or 2-$MRMP$) the maximum related multiset partition of S and T, and by $|MRMP(S, T)|$ (or $|2$-$MRMP|$) the size of the partition, *i.e.*, the number of related multiset pairs in the partition.

LEMMA 2.5. *Given a common integer partition $CIP(S, T)$, we can transform it into a related multiset partition of S and T, denoted as $RMP(S, T)$, such that $|RMP(S, T)| \geq |S| + |T| - |CIP(S, T)|$.*

PROOF. Based on the given common integer partition $CIP(S, T)$, $\langle S, T \rangle$ can be decomposed into l connected components $\langle S_1, T_1 \rangle$, $\langle S_2, T_2 \rangle$, $\cdots$, $\langle S_l, T_l \rangle$. By Lemma 2.4, each connected component is a pair of related multisets and disjoint with any other component. Therefore, all the l connected components naturally give a related multiset partition (denoted as $RMP(S, T)$) of $\langle S, T \rangle$, such that $|RMP(S, T)| = l$. Let $CIP(S_i, T_i)$ denote the common integer partition of $\langle S_i, T_i \rangle$ induced from $CIP(S, T)$. We can see that the union of all $CIP(S_i, T_i)$ will be the common integer partition $CIP(S, T)$, *i.e.*, $CIP(S, T) = \uplus_{i=1}^{l} CIP(S_i, T_i)$. Since each $\langle S_i, T_i \rangle$ is a connected component, by Theorem 2.4, $|CIP(S_i, T_i)| \geq |S_i| + |T_i| - 1$ holds for each $i \in [1, l]$. Therefore, $|CIP(S, T)| = \sum_{i=1}^{l} |CIP(S_i, T_i)| \geq \sum_{i=1}^{l} (|S_i| + |T_i| - 1)| = |S| + |T| - |RMP(S, T)|$. $\square$

The following lemma establishes the relationship between MCIP and MRMP,

showing their (complementary) equivalence.

LEMMA 2.6. *If S and T are two related multisets, then $|MCIP(S,T)|+|MRMP(S,T)| = |S|+|T|$.*

PROOF. Assume that $\langle S_1, T_1\rangle$, $\langle S_2, T_2\rangle$, $\cdots$, $\langle S_l, T_l\rangle$ is a maximum related multiset partition of S and T with $l = |MRMP(S,T)|$. For each $i \in [1, l]$, $\langle S_i, T_i\rangle$ is a pair of basic related multisets, and by Lemma 2.2, the minimum common integer partition $MCIP(S_i, T_i)$ is of size less than or equal to $|S_i|+|T_i|-1$, *i.e.*, $|MCIP(S_i, T_i)| \leq |S_i| + |T_i| - 1$. We can also see that the union of all $MCIP(S_i, T_i)$ forms a common integer partition $CIP(S,T)$ of S and T, *i.e.*, $CIP(S,T) = \biguplus_{i=1}^{l} MCIP(S_i, T_i)$, and its size is $|CIP(S,T)| = \sum_{i=1}^{l} |MCIP(S_i, T_i)| \leq \sum_{i=1}^{l}(|S_i| + |T_i| - 1)| = |S| + |T| - |MRMP(S,T)|$. Therefore, we have $|MCIP(S,T)| \leq |CIP(S,T)| \leq |S| + |T| - |MRMP(S,T)|$.

By Lemma 2.5, given a minimum common integer partition $MCIP(S,T)$, we can transform it into a related multiset partition $RMP(S,T)$ such that $|RMP(S,T)| \geq |S| + |T| - |MCIP(S,T)|$. Because $|MRMP(S,T)| \geq |RMP(S,T)|$, we have $|MCIP(S,T)| \geq |S| + |T| - |MRMP(S,T)|$. $\square$

Since a pair of basic related multisets S and T cannot be partitioned further into pairs of related submultisets, *i.e.*, $|MRMP(S,T)| = 1$, the following lemma is trivially implied by Lemma 2.6.

LEMMA 2.7. *If S and T are a pair of basic related multisets, then $|MCIP(S,T)| = |S| + |T| - 1$.*

The following lemmas will be crucial to the approximation algorithms. We define the size of a pair of related multisets S and T as the sum of the size of S and the size of T, *i.e.*, $|\langle S, T\rangle| = |S| + |T|$.

LEMMA 2.8. *If the minimum size of any pair of related submultisets of S and T is c, then $|MCIP(S,T)| \geq \frac{c-1}{c}(|S| + |T|)$.*

PROOF. Assume that $\{\langle S_1, T_1\rangle, \langle S_2, T_2\rangle, \cdots, \langle S_l, T_l\rangle\}$ are the pairs of basic related multisets induced by the minimum common integer partition $MCIP(S,T)$, such that $|MCIP(S,T)| = \sum_{i=1}^{l}(|S_i|+|T_i|-1)$. Since $|\langle S_i, T_i\rangle| = |S_i|+|T_i| \geq c$ for each $i \in [1, n]$, we have $|MCIP(S,T)| = \sum_{i=1}^{l}(|S_i|+|T_i|-1) = \sum_{i=1}^{l} \frac{|S_i|+|T_i|-1}{|S_i|+|T_i|}(|S_i|+|T_i|) \geq \sum_{i=1}^{l}(1-\frac{1}{c})(|S_i|+|T_i|) = \frac{c-1}{c}(|S|+|T|)$. $\square$

LEMMA 2.9. *Given two related multisets, $S = \{x_1, x_2, \cdots, x_m\}$ and $T = \{y_1, y_2, \cdots, y_n\}$. If x_i and y_j are a pair of identical integers, then $\{x_i\} \uplus MCIP(S\backslash\{x_i\}, T\backslash \{y_j\})$ is a minimum common integer partition of S and T, i.e., $|MCIP(S,T)| = |MCIP(S \setminus \{x_i\}, T \setminus \{y_j\})| + 1$.*

PROOF. Assume that $MCIP(S,T)$ is a minimum common integer partition of S and T. Let $MRMP(S,T)$ denote the maximum related multiset partition induced by $MCIP(S,T)$. If x_i and y_j are in the same pair of related submultisets $\langle S_1, T_1\rangle$ of $MRMP(S,T)$ such that $\langle S_1, T_1\rangle \neq \langle\{x_i\}, \{y_j\}\rangle$, then we can further decompose $\langle S_1, T_1\rangle$ into two pairs of related submultisets $\langle\{x_i\}, \{y_j\}\rangle$ and $\langle S_1 \setminus \{x_i\}, T_1 \setminus \{y_j\}\rangle$, which contradicts the definition of the maximum related multiset partition. If x_i and y_j are in two pairs of distinct related submultisets $\langle S_1, T_1\rangle$

Algorithm $\frac{5}{4}$-Approx-MCIP(S, T)

input Two related multisets S and T
output A common integer partition CIP of S and T
begin
    remove_common_integer(S,T);
    approximate_set_packing(S,T);
    $CIP := CIP(S_1, T_1) \uplus CIP(S_2, T_2)$;
    $CIP := CIP \uplus$ 2-APPROX-MCIP(S_3, T_3);
    return CIP;
end.

Fig. 2.   A $\frac{5}{4}$-approximation algorithm for 2-MCIP.

and $\langle S_2, T_2 \rangle$ of $MRMP(S,T)$, respectively, then we can obtain a new maximum related multiset partition by replacing $\langle S_1, T_1 \rangle$ and $\langle S_2, T_2 \rangle$ with $\langle \{x_i\}, \{y_j\} \rangle$ and $\langle S_1 \uplus S_2 \setminus \{x_i\}, T_1 \uplus T_2 \setminus \{y_j\} \rangle$. Moreover, by Lemma 2.6, the new maximum related multiset partition gives another minimum common integer partition in which x_i is mapped to y_j, implying that $|MCIP(S,T)| = |MCIP(S \setminus \{x_i\}, T \setminus \{y_j\})| + 1$. $\square$

Unfortunately, the result in Lemma 2.9 cannot be extended to the case of k multisets when $k \geq 3$. An interesting counterexample is $\{6, 5, 1, 4, 2\}$, $\{6, 5, 1, 3, 3\}$, $\{6, 4, 2, 3, 3\}$. Their minimum common integer partition is of size 6, but any common integer partition including 6 as an element is of size at least 7. In the following, we will use a procedure **remove_common_integer**($S_1, S_2, \cdots, S_k$) to remove all common integer elements existing in every multiset of $\{S_1, S_2, \cdots, S_k\}$ (and add them into the solution). The optimality of this operation is guaranteed only when $k = 2$, as shown in Lemma 2.9.

3. APPROXIMATION OF 2-MCIP VIA MAXIMUM SET PACKING

In this section, we will give a $\frac{5}{4}$-approximation algorithm for the 2-MCIP problem by considering pairs of basic related submultisets of sizes three and four between S and T. As mentioned earlier (see Lemma 2.9), we assume that there are no common integer elements between the two input multisets S and T, without loss of generality.

We can construct an instance of the *Maximum Set Packing* problem [Ausiello et al. 1999], in which the collection C consists of all pairs of basic related submultisets of sizes three and four between S and T. Note that, each pair of basic related multisets can also be viewed as a multiset. Since the cardinality of each such pair (or multiset) in C is bounded from the above by a constant, it is actually an instance of the *Maximum k-Set Packing* problem where $k = 4$. Hurkens and Schrijver [1989] show that the Maximum k-Set Packing problem is approximable within ratio $k/2 + \epsilon$ for any $\epsilon > 0$. For the weighted version of the Maximum k-Set Packing problem, where each set is given a non-negative weight, Arkin and Hassin [1998] show that it is approximable within ratio $k - 1 + \epsilon$ for any $\epsilon > 0$.

In the following, we consider a special weighted Maximum k-Set Packing problem on C, where the weight for each pair of basic related multisets of size three (which is simply considered as a multiset of size three) is 2 and the weight for a pair of basic

related multisets of size four (which is considered as a multiset of size four) is 1, and the goal is to find a collection of *disjoint* multisets of the maximum total weight. Call any collection of pairwise disjoint multisets a *packing*. We design a heuristic algorithm, which is implemented in the procedure **approximate_set_packing**(S,T), to find a packing as follows: first find a *maximal* set packing, and then recursively replace a multiset of size four in the packing by a multiset of size three, or replace a multiset of size three by two multisets of size three, or add some multiset into the packing so that the resultant collection is still a packing (but with one more multiset of size three after a replacement or with one more multiset after an addition), until no such replacement or addition could be made further.

The above heuristic algorithm can be made to run in $O(|U| \cdot |C|^2)$ time, where U denotes the universe of the elements in Set Packing, *i.e.*, the multiset union of all multisets in C. In our case, $|U| \leq m + n$. To see this running time, first, given a packing P, we define a mapping $f_P : U \mapsto C \cup \{\emptyset\}$ as follows: for $\forall u \in U$, $f_P(u) = c$ if there exists a multiset $c \in P$ such that $u \in c$, and $f_P(u) = \emptyset$ otherwise. Second, given a multiset $c \in C$, the multisets in the packing P that are not disjoint with c can be found in constant time by looking up the mapping function f_P, because c has only three or four elements. Furthermore, given two multisets c_1 and c_2 in C, we can find in constant time a multiset p in P such that if p is replaced by c_1 and c_2 then $P' = P \uplus \{c_1, c_2\} \setminus \{p\}$ is still a packing of C, or report no such a multiset p in P exists. Third, after each replacement or addition, updating the mapping function for P' can also be done with f_P in constant time. Therefore, in our heuristic algorithm, a replacement or addition at each iteration can be made in $|C|^2$ time as we may enumerate every pair of multisets in C for a possible replacement or addition. Finally, observe that at most $|U|$ replacements could be made as the number of multisets of size three in the found packing increases by one after each replacement; also observe that at most $|U|$ additions could be made as the number of multisets in a maximal set packing is at most $|U|$.

Let q_3 and q_4 denote the numbers of pairs of basic related multisets of sizes three and four in the packing found by our heuristic algorithm, and q_3^* and q_4^* the numbers of pairs of basic related multisets of sizes three and four in an optimal weighted set packing, respectively. It is obvious that $2q_3 + q_4 \leq 2q_3^* + q_4^*$. Moreover, we can obtain the following relationship. [2]

LEMMA 3.1. $2q_3^* + q_4^* \leq 4(q_3 + q_4)$.

PROOF. Let $Q_{i,j}^*$, where $i \in \{3,4\}$ and $1 \leq j \leq i$, be a collection of multisets of size i in the optimal set packing that intersect j multisets in the packing found by our heuristic algorithm, and $q_{i,j}^*$ be the cardinality of $Q_{i,j}^*$. Because the packing found by our heuristic is maximal, we can see that $q_3^* = \sum_{j=1}^{3} q_{3,j}^*$ and $q_4^* = \sum_{j=1}^{4} q_{4,j}^*$. Observe that every multiset of size three (and four) in the packing found by the heuristic intersects at most three (and four, respectively) multisets in the optimal packing, which implies that $\sum_{j=1}^{3} j \cdot q_{3,j}^* + \sum_{j=1}^{4} j \cdot q_{4,j}^* \leq 3q_3 + 4q_4$. Furthermore, no two multisets in $Q_{3,1}^*$ can intersect a same multiset in the packing

---

[2] The $(k/2 + \epsilon)$-approximation algorithm given by Hurkens and Schrijver [Hurkens and Schrijver 1989] can also find a packing of C satisfying the inequality in Lemma 3.1, but only in quasi-polynomial time.

of the heuristic and none of multisets in $Q_{3,1}^*$ intersect a multiset of size four in the packing of the heuristic either, implying that $q_{3,1}^* \leq q_3$. Therefore, it follows that $2q_3^* + q_4^* = 2\sum_{j=1}^3 q_{3,j}^* + \sum_{j=1}^4 q_{4,j}^* \leq q_{3,1}^* + (\sum_{j=1}^3 j \cdot q_{3,j}^* + \sum_{j=1}^4 j \cdot q_{4,j}^*) \leq 4(q_3 + q_4)$. □

Let q_3' and q_4' be the numbers of pairs of basic related submultisets of sizes three and four in the related multiset partition induced by a given minimum common partition $\mathrm{MCIP}(S, T)$. It is obvious that $2q_3' + q_4' \leq 2q_3^* + q_4^*$. The following is a tighter lower bound for 2-MCIP.

LEMMA 3.2. $|MCIP(S, T)| \geq \frac{4}{5}(m + n) - \frac{1}{5}(2q_3^* + q_4^*)$, where $m = |S|$ and $n = |T|$.

PROOF. Based on the given minimum common integer partition $\mathrm{MCIP}(S, T)$, we can partition $\langle S, T \rangle$ into three pairs of disjoint related submultisets: $\langle S_1, T_1 \rangle$, which consists of the integer elements in the pairs of basic related submultisets of size three; $\langle S_2, T_2 \rangle$, which consists of the integer elements in the pairs of basic related submultisets of size four; and $\langle S_3, T_3 \rangle$, which includes the remaining elements in $\langle S, T \rangle$ such that, $S = S_1 \uplus S_2 \uplus S_3$ and $T = T_1 \uplus T_2 \uplus T_3$. Therefore, we have

$$\begin{aligned} |MCIP(S, T)| &= MCIP(S_1, T_1)| + |MCIP(S_2, T_2)| + |MCIP(S_3, T_3)| \\ &= 2q_3' + 3q_4' + |MCIP(S_3, T_3)| \end{aligned}$$

Since each pair of the basic related submultisets of $\langle S_3, T_3 \rangle$ induced by $\mathrm{MCIP}(S, T)$ is of size at least five, by Lemma 2.8, we have

$$|MCIP(S_3, T_3)| \geq \frac{4}{5}(m + n - 3q_3' - 4q_4')$$

and thus

$$|MCIP(S, T)| \geq \frac{4}{5}(m + n) - \frac{1}{5}(2q_3' + q_4') \geq \frac{4}{5}(m + n) - \frac{1}{5}(2q_3^* + q_4^*)$$

from which the lemma follows. □

The following lemma gives a tighter upper bound for 2-MCIP.

LEMMA 3.3. $|MCIP(S, T)| \leq m + n - q_3 - q_4 - 1$.

PROOF. Observe that we can partition $\langle S, T \rangle$ into three pairs of disjoint related submultisets: $\langle S_1, T_1 \rangle$, which consists of the integer elements in the q_3 pairs of basic related submultisets of size three; $\langle S_2, T_2 \rangle$, which consists of the integer elements in the q_4 pairs of basic related submultisets of size four; and $\langle S_3, T_3 \rangle$, which includes the remaining elements in $\langle S, T \rangle$, i.e., $S = S_1 \uplus S_2 \uplus S_3$ and $T = T_1 \uplus T_2 \uplus T_3$. Therefore, we have

$$\begin{aligned} |MCIP(S, T)| &\leq |MCIP(S_1, T_1)| + |MCIP(S_2, T_2)| + |MCIP(S_3, T_3)| \\ &\leq 2q_3 + 3q_4 + |MCIP(S_3, T_3)| \end{aligned}$$

Moreover, by Lemma 2.2 we have

$$|MCIP(S_3, T_3)| \leq m + n - 3q_3 - 4q_4 - 1$$

and thus

$$|MCIP(S, T)| \leq m + n - q_3 - q_4 - 1$$

```
Algorithm k-APPROX-MCIP(S_1, S_2,···, S_k)

input A sequence of related multisets S_1, S_2,···, S_k
output A common integer partition CIP of S_1, S_2,···, S_k
begin
    CIP := 2-APPROX-MCIP(S_1, S_2);
    for i = 3 to k do
        CIP := 2-APPROX-MCIP(CIP, S_i);
    return CIP;
end.
```

Fig. 3. A k-approximation algorithm for k-MCIP.

from which the lemma follows. □

As mentioned earlier, we run the procedure **approximate_set_packing**(S,T) to find the three disjoint submultisets $\langle S_1, T_1 \rangle$, $\langle S_2, T_2 \rangle$ and $\langle S_3, T_3 \rangle$. A $\frac{5}{4}$-approximation algorithm for 2-MCIP can then be obtained, as illustrated in Figure 2. The algorithm runs in time $O((m + n)^9)$, which is dominated by the running time of the procedure **approximate_set_packing**(S,T), as there are $m + n$ elements in the universe and the size of the collection C could reach $\Theta((m + n)^4)$ in the worst case. We believe that the running time can be further reduced by a more careful implementation and analysis of the procedure **approximate_set_packing**(S,T).

THEOREM 3.4. *The algorithm* $\frac{5}{4}$*-APPROX-MCIP is a* $\frac{5}{4}$*-approximation algorithm for 2-MCIP.*

PROOF. By Lemmas 3.2 and 3.3, the approximation ratio α given by algorithm $\frac{5}{4}$-APPROX-MCIP is

$$\alpha \leq \frac{m + n - q_3 - q_4 - 1}{\frac{4}{5}(m + n) - \frac{1}{5}(2q_3^* + q_4^*)} = \frac{5}{4} \cdot \frac{m + n - q_3 - q_4 - 1}{m + n - \frac{1}{4}(2q_3^* + q_4^*)}$$

It suffices to show that $m + n - q_3 - q_4 - 1 \leq m + n - \frac{1}{4}(2q_3^* + q_4^*)$, which is equivalent to showing $2q_3^* + q_4^* \leq 4(q_3 + q_4 + 1)$. By Lemma 3.1, we know that $2q_3^* + q_4^* \leq 4(q_3 + q_4)$. Therefore, $\alpha \leq \frac{5}{4}$. □

4. APPROXIMATION OF K-MCIP

In this section, we will discuss how to approximate the general k-MCIP ($k \geq 3$) problem.

Using the algorithm 2-Approx-MCIP(S,T) in the previous section, we give an approximation algorithm to solve the k-MCIP ($k \geq 3$) problem, as described in Figure 3. First, we give an upper bound on the performance of this algorithm.

LEMMA 4.1. $|MCIP(S_1, S_2, \cdots, S_k)| \leq \sum_{i=1}^{k} |S_i| - k + 1$.

PROOF. After the multiset S_j is processed in the algorithm k-Approx-MCIP$(S_1, S_2, \cdots, S_k)$, by Lemma 2.2, the size of the common integer partition found so far is upper bounded by $\sum_{i=1}^{j} |S_i| - j + 1$, which holds until j increases up to k. □

THEOREM 4.2. *The algorithm* k-APPROX-MCIP *is a* k-*approximation algorithm for the* k-*MCIP* ($k \geq 2$) *problem.*

Algorithm $\frac{3k(k-1)}{3k-2}$-Approx-MCIP$(S_1, S_2, \cdots, S_k)$

input A sequence of related multisets $S_1, S_2, \cdots, S_k$
output A common integer partition CIP of $S_1, S_2, \cdots, S_k$
begin
    remove_common_integer$(S_1, S_2, \cdots, S_k)$;
    $CIP := k\text{-Approx-MCIP}(S_1, S_2, \cdots, S_k)$;
    return CIP;
end.

Fig. 4.   A $\frac{3k(k-1)}{3k-2}$-approximation algorithm for k-MCIP.

PROOF. By Lemma 2.1 and Lemma 4.1, the size of the common integer partition CIP returned from k-Approx-MCIP$(S_1, S_2, \cdots, S_k)$ is such that $max\{|S_1|, |S_2|, \cdots, |S_k|\} \leq |MCIP(S_1, S_2, \cdots, S_k)| \leq |CIP(S_1, S_2, \cdots, S_k)| \leq \sum_{i=1}^{k} |S_i| - k + 1$, from which the theorem follows. □

As described in Figure 4, the algorithm k-Approx-MCIP can be slightly improved by employing the procedure **remove_common_integer**$(S_1, S_2, \cdots, S_k)$. To show that this improved algorithm achieves an approximation ratio less than k, we need the following lemma.

LEMMA 4.3. *If there is no integer element common to all the multisets in $\{S_1, S_2, \cdots, S_k\}$, then it holds that $|MCIP(S_1, S_2, \cdots, S_k)| \geq \frac{3k-2}{3k(k-1)} \sum_{i=1}^{k} |S_i|$.*

PROOF. We can see that, there is always a multiset among $S_1, S_2, \cdots, S_k$ such that its size is no less than $\frac{1}{k} \sum_{i=1}^{k} |S_i|$. Without loss of generality, we assume that this multiset is S_k. In an optimal solution MCIP, with respect to S_k, we can divide the elements in a multiset $S_i (1 \leq i \leq k-1)$ into two disjoint submultisets: S_i^1, which consists of elements that are mapped to exactly one (identical) integer in S_k; and S_i^2, which is the complement submultiset of S_i^1, *i.e.*, $S_i^2 = S_i \setminus S_i^1$. Accordingly, with respect to any S_i, S_k can be divided into two disjoint multisets: $S_{k,i}^1$, which consists of elements that are mapped to an integer in S_i^1; and $S_{k,i}^2$, which is the complement submultiset of $S_{k,i}^1$, *i.e.*, $S_{k,i}^2 = S_k \setminus S_{k,i}^1$. Obviously, S_i^1 and $S_{k,i}^1$ are a pair of related multisets, as are S_i^2 and $S_{k,i}^2$.

Notice that, we can always choose a particular multiset S_j, where $1 \leq j \leq k-1$, such that $2|S_j| - |S_j^1| \geq \frac{1}{k-1} \sum_{i=1}^{k-1} (2|S_i| - |S_i^1|)$. In addition, $\sum_{i=1}^{k-1} |S_i^1| \leq (k-2)|S_k|$ holds because $S_1, S_2, \cdots, S_k$ have no common elements. We have

$$|MCIP(S_1, S_2, \cdots, S_k)|$$
$$\geq \frac{2}{3}(|S_k| + |S_j| - 2|S_j^1|) + |S_j^1| \quad \text{(By Lemma 2.8)}$$
$$= \frac{1}{3}\{2|S_k| + 2|S_j| - |S_j^1|\}$$
$$\geq \frac{1}{3}\{2|S_k| + \frac{1}{k-1} \sum_{i=1}^{k-1} (2|S_i| - |S_i^1|)\} \quad (2|S_j| - |S_j^1| \geq \frac{1}{k-1} \sum_{i=1}^{k-1}(2|S_i| - |S_i^1|))$$

$$= \frac{1}{3}\{2|S_k| + \frac{1}{k-1}(2\sum_{i=1}^{k-1}|S_i| - \sum_{i=1}^{k-1}|S_i^1|)\}$$

$$\geq \frac{1}{3}\{2|S_k| + \frac{1}{k-1}(2\sum_{i=1}^{k-1}|S_i| - (k-2)|S_k|)\} \quad (\sum_{i=1}^{k-1}|S_i^1| \leq (k-2)|S_k|)$$

$$= \frac{1}{3}\{\frac{k}{k-1}|S_k| + \frac{2}{k-1}\sum_{i=1}^{k-1}|S_i|\}$$

$$= \frac{1}{3}\{\frac{k-2}{k-1}|S_k| + \frac{2}{k-1}\sum_{i=1}^{k}|S_i|\}$$

$$\geq \frac{1}{3}\{\frac{k-2}{k-1} \cdot \frac{\sum_{i=1}^{k}|S_i|}{k} + \frac{2}{k-1}\sum_{i=1}^{k}|S_i|\} \quad (|S_k| \geq \frac{1}{k}\sum_{1}^{k}|S_i|)$$

$$= \frac{3k-2}{3k(k-1)}\sum_{i=1}^{k}|S_i|$$

□

THEOREM 4.4. *The algorithm* $\frac{3k(k-1)}{3k-2}$*-*APPROX-MCIP *is a* $\frac{3k(k-1)}{3k-2}$*-approximation algorithm for the k-MCIP (k ≥ 2) problem.*

PROOF. We consider lower and upper bounds of the size of $MCIP(S_1, S_2, \cdots, S_k)$. Let q denote the number of common integers of $\{S_1, S_2, \cdots, S_k\}$ used in a given minimum common integer partition $MCIP(S_1, S_2, \cdots, S_k)$, and q^* the maximum number of common integers, which is always returned by the procedure **remove_ common_integer**$(S_1, S_2, \cdots, S_k)$. Obviously, $q \leq q^*$. By Lemma 4.3, we have

$$|MCIP(S_1, S_2, \cdots, S_k)| \geq q + \frac{3k-2}{3k(k-1)}\sum_{i=1}^{k}(|S_i| - q)$$

On the other hand, it follows from the definition of the algorithm $\frac{3k(k-1)}{3k-2}$-APPROX-MCIP and Lemma 4.1 that

$$|MCIP(S_1, S_2, \cdots, S_k)| \leq q^* + \sum_{i=1}^{k}(|S_i| - q^*)$$

Therefore, the approximation ratio α achieved by the algorithm $\frac{3k(k-1)}{3k-2}$-APPROX-MCIP is

$$\alpha \leq \frac{q^* + \sum_{i=1}^{k}(|S_i| - q^*)}{q + \frac{3k-2}{3k(k-1)}\sum_{i=1}^{k}(|S_i| - q)} = \frac{3k(k-1)}{3k-2} \cdot \frac{\sum_{i=1}^{k}|S_i| - (k-1)q^*}{\sum_{i=1}^{k}|S_i| - \frac{k}{3k-2}q}$$

Now we show that $(k-1)q^* \geq \frac{k}{3k-2}q$. Since $q \leq q^*$, it is sufficient to prove that $k - 1 \geq \frac{k}{3k-2}$, which is obvious to hold for any $k \geq 2$. □

Clearly, the algorithm $\frac{3k(k-1)}{3k-2}$-APPROX-MCIP$(S_1, S_2, \cdots, S_k)$ runs in $O(\sum_{i=1}^{k}|S_i| \cdot log(\sum_{i=1}^{k}|S_i|))$ time. Let us compare Theorem 4.4 with Theorem 4.2. Clearly,

$\frac{3k(k-1)}{3k-2}$ is always smaller than k, for any $k \geq 2$. For example, when $k = 2$, the above algorithm gives approximation ratio 1.5, and when $k = 3$, its approximation ratio is $\frac{18}{7}$, which is much better than the ratio 3 in Theorem 4.2. However, when k becomes large, $\frac{3k(k-1)}{3k-2}$ is only slightly smaller than k, since $\frac{3k(k-1)}{3k-2} = \Theta(k)$. It is an interesting open question whether k-MCIP has an approximation algorithm with a ratio that is asymptotically better than k.

5. HARDNESS OF APPROXIMATION

It is easy to see that MCIP is NP-hard because there is a straightforward reduction from the *Subset Sum* problem. This section is devoted to proving that MCIP is APX-hard.

In the sequel, we prove the APX-completeness of 2-MCIP by an L-reduction from the *Maximum Bounded 3-Dimensional Matching* problem (denoted as MAX 3DM-3). The MAX 3DM-3 problem is defined as follows: given a set $D \subseteq X \times Y \times Z$, where X, Y and Z are disjoint sets and moreover, each element in X, Y and Z occurs in at least one and at most three triples in D [Kann 1991], the goal is to find a matching $M \subseteq D$ for D of the maximum cardinality, *i.e.*, a largest set $M \subseteq D$ such that no two elements in M agree in any coordinate. In this problem, without loss of generality, we can assume that $n = |X| \leq |Y| \leq |Z|$. Since each element in X occurs at least once and at most three times in D, the number of triples is at least n and at most $3n$, *i.e.*, $n \leq |D| \leq 3n$. It also implies that $|Y| \leq 3n$ and $|Z| \leq 3n$. Further observe that each triple can intersect at most six other triples, which implies that the maximum matching contains at least $|D|/7$ triples. Let $|MAX\ 3DM\text{-}3|$ denote the size of maximum matching of $|D|$. It is easy to see that $\lceil \frac{n}{7} \rceil \leq |MAX\ 3DM\text{-}3| \leq n$.

Let $X = \{x_1, x_2, \cdots, x_{|X|}\}$, $Y = \{y_1, y_2, \cdots, y_{|Y|}\}$, $Z = \{z_1, z_2, \cdots, z_{|Z|}\}$, and $D = \{d_1, d_2, \cdots, d_{|D|}\}$ where $d_i = (x_{i^X}, y_{i^Y}, z_{i^Z})$ for each $i \in [1, |D|]$ and i^X (i^Y or i^Z, respectively) is the corresponding index of the integer x_{i^X} (y_{i^Y} or z_{i^Z}, respectively) in X (Y or Z, respectively). We can define a function f to construct an instance of 2-MCIP as follows:

(1) A multiset $\tilde{X} = \{\tilde{x}_i | \tilde{x}_i = 4^i, \forall x_i \in X\}$;

(2) A multiset $\tilde{Y} = \{\tilde{y}_i | \tilde{y}_i = 4^{|X|+i}, \forall y_i \in Y\}$;

(3) A multiset $\tilde{Z} = \{\tilde{z}_i | \tilde{z}_i = 4^{|X|+|Y|+i}, \forall z_i \in Z\}$;

(4) A multiset $\tilde{D} = \{\tilde{d}_i | \tilde{d}_i = \tilde{x}_{i^X} + \tilde{y}_{i^Y} + \tilde{z}_{i^Z}, \forall d_i \in D\}$;

(5) An integer $e = \sum_{i=1}^{|D|} \tilde{d}_i - \sum_{i=1}^{|X|} \tilde{x}_i - \sum_{i=1}^{|Y|} \tilde{y}_i - \sum_{i=1}^{|Z|} \tilde{z}_i$.

(6) Two multisets $S = \tilde{D}$ and $T = \tilde{X} \cup \tilde{Y} \cup \tilde{Z} \cup \{e\}$.

Since each element in X, Y and Z is assumed to occur at least once in D while some elements occur more than once, it always holds that $e > 0$. Obviously, $\sum S = \sum T$. Therefore, $\langle S, T \rangle$ is an instance of 2-MCIP that we can obtain in time linear in n.

Let $|2\text{-}MCIP|$ denote the size of the minimum common integer partition of $\langle S, T \rangle$. Then, we have the following lemma.

LEMMA 5.1. *For any instance of MAX 3DM-3, $|2\text{-}MCIP| \leq 70 \cdot |MAX\ 3DM\text{-}3|$.*

PROOF. By Lemma 2.2, we have $|2\text{-}MCIP| \leq |X| + |Y| + |Z| + |D| \leq 10n$. On the other hand, we have shown that $|MAX\ 3DM\text{-}3| \geq \lceil \frac{n}{7} \rceil$. The lemma thus follows. $\square$

Given a common integer partition $2\text{-}CIP$ of $\langle S,T \rangle$, we define a function g to construct a subset (denoted as $3DM\text{-}3$) of D by including all the triples $d_i = (x_{i^X}, y_{i^Y}, z_{i^Z})$ $(1 \leq i \leq |D|)$ whose corresponding integers $\tilde{d}_i = \tilde{x}_{i^X} + \tilde{y}_{i^Y} + \tilde{z}_{i^Z}$ are not connected to the integer e in the common integer partition $2\text{-}CIP$.

LEMMA 5.2. *For any instance D of MAX 3DM-3, the subset $3DM$-3 constructed by the function g is a matching of D.*

PROOF. Let $S_1 = \{\tilde{d}_{\varepsilon_1}, \cdots, \tilde{d}_{\varepsilon_k}\}$ and $T_1 = \{\tilde{x}_{\chi_1}, \cdots, \tilde{x}_{\chi_l}, \tilde{y}_{\gamma_1}, \cdots, \tilde{y}_{\gamma_m}, \tilde{z}_{\zeta_1}, \cdots, \tilde{z}_{\zeta_n}\}$ include all the integers that are not connected to e in the given common integer partition $2\text{-}CIP$ of S and T. It can be seen that, $3DM\text{-}3 = \{d_{\varepsilon_1}, \cdots, d_{\varepsilon_k}\}$, and $\langle S_1, T_1 \rangle$ are a pair of related multisets, *i.e.*,

$$\sum_{i=1}^{k} \tilde{d}_{\varepsilon_i} = \sum_{i=1}^{l} \tilde{x}_{\chi_i} + \sum_{i=1}^{m} \tilde{y}_{\gamma_i} + \sum_{i=1}^{n} \tilde{z}_{\zeta_i}$$

By definition, $\tilde{d}_{\varepsilon_i} = \tilde{x}_{\varepsilon_i^X} + \tilde{y}_{\varepsilon_i^Y} + \tilde{z}_{\varepsilon_i^Z}$, for each $i \in [1, k]$. Thus,

$$\sum_{i=1}^{k} \tilde{x}_{\varepsilon_i^X} + \sum_{i=1}^{k} \tilde{y}_{\varepsilon_i^Y} + \sum_{i=1}^{k} \tilde{z}_{\varepsilon_i^Z} = \sum_{i=1}^{l} \tilde{x}_{\chi_i} + \sum_{i=1}^{m} \tilde{y}_{\gamma_i} + \sum_{i=1}^{n} \tilde{z}_{\zeta_i} \qquad (1)$$

In order to prove that $3DM\text{-}3$ is a matching of D, it is sufficient to show that the following three pairs of index sets are identical: $\{\varepsilon_1^X, \cdots, \varepsilon_k^X\} = \{\chi_1, \cdots, \chi_l\}$, $\{\varepsilon_1^Y, \cdots, \varepsilon_k^Y\} = \{\gamma_1, \cdots, \gamma_m\}$ and $\{\varepsilon_1^Z, \cdots, \varepsilon_k^Z\} = \{\zeta_1, \cdots, \zeta_n\}$, since no integer element has two copies in T_1. Also notice that, by definition, no two integer elements in T are of equal value.

Let us first assume that $\tilde{x}_{\chi_1}(= 4^{\chi_1})$ be the smallest integer in $S_1 \uplus T_1$, and apply mod 4^{χ_1+1} to Equation (1); that is,

$$\text{Equation (1)} \qquad \text{mod } 4^{\chi_1+1}$$

$$\Rightarrow \sum_{i=1}^{k} \tilde{x}_{\varepsilon_i^X} \equiv \tilde{x}_{\chi_i} \quad \text{mod } 4^{\chi_1+1}$$

This is because any integer in S_1 or T_1 rather than $\tilde{x}_{\chi_1}$ is divisible by 4^{χ_1+1}. On the other hand, the integer $\tilde{x}_{\chi_1}$ occurs at most three times in the multiset $\{\tilde{x}_{\varepsilon_i^X} | 1 \leq i \leq k\}$, and the base that we use to define the integers in S and T is four. Therefore, the above modulo equivalence implies that there is exactly one integer element of $\tilde{x}_{\chi_1}$ in S_1.

If the smallest integer in $S_1 \uplus T_1$ is $\tilde{x}_{\varepsilon_1^X}$, we can use the same arguments as above to show that there is exactly one integer element of $\tilde{x}_{\varepsilon_1^X}$ in S_1 and also in T_1. Therefore, we can remove the smallest integer from S_1 and T_1, and then repeat the above procedure until the three pairs of index sets are shown to be identical. $\square$

Let $|2\text{-}MRMP|$ be the size of the maximum related multiset partition of S and T.

LEMMA 5.3. $|2\text{-}MRMP| = |MAX\ 3DM\text{-}3| + 1$.

PROOF. We can see that, each triple $d_i = (x_{i^X}, y_{i^Y}, z_{i^Z})$ in a maximum matching naturally leads to a pair of related submultisets, *i.e.*, $\{\tilde{d}_i\}$ and $\{\tilde{x}_{i^X}, \tilde{y}_{i^Y}, \tilde{z}_{i^Z}\}$. In addition, there is a pair of related submultisets that contain the integer e. Therefore, $|2\text{-}MRMP| \geq |MAX\ 3DM\text{-}3| + 1$.

In a maximum related multiset partition, there is only one pair of related submultisets that contain the integer e. In any other pair of related submultisets, there exists at least one integer $\tilde{d}_i$, whose corresponding triple d_i will be included in a matching of D. Therefore, $|2\text{-}MRMP| - 1 \leq |MAX\ 3DM\text{-}3|$. □

Let $|2\text{-}RMP|$ be the size of a related multiset partition of S and T, induced by a given common partition 2-CIP.

LEMMA 5.4. $|MAX\ 3DM\text{-}3| - |3DM\text{-}3| \leq |2\text{-}CIP| - |2\text{-}MCIP|$.

PROOF. We have shown in the proof of the previous lemma that, given a pair of related submultisets that does not include e, there exists at least one integer $\tilde{d}_i$, whose corresponding triple d_i will be included in a matching of D. Therefore, $|2\text{-}RMP| - 1 \leq |3DM\text{-}3|$. By Lemma 5.3, we have $|MAX\ 3DM\text{-}3| - |3DM\text{-}3| \leq |2\text{-}MRMP| - |2\text{-}RMP|$. On the other hand, $|2\text{-}MCIP| + |2\text{-}MRMP| = |X| + |Y| + |Z| + |D| + 1$ by lemma 2.6, and $|2\text{-}CIP| + |2\text{-}RMP| \geq |X| + |Y| + |Z| + |D| + 1$ by Lemma 2.5. Therefore, $|2\text{-}MRMP| - |2\text{-}RMP| \leq |2\text{-}CIP| - |2\text{-}MCIP|$, from which the lemma follows. □

LEMMA 5.5. $MAX\ 3DM\text{-}3 \leq_L 2\text{-}MCIP$.

PROOF. By Lemmas 5.1 and 5.4, the quadruple $(f, g, 70, 1)$ discussed above gives an L-reduction from MAX 3DM-3 to the 2-MCIP problem [Papadimitriou and Yannakakis 1991]. □

THEOREM 5.6. *The k-MCIP problem is APX-complete, for any $k \geq 2$.*

PROOF. Since the MAX 3DM-3 problem is APX-complete [Kann 1991] and MAX 3DM-3 $\leq_L$ 2-MCIP by Lemma 5.5, 2-MCIP is APX-hard. In addition, by Lemma 2.3, there exists a polynomial-time 2-approximation algorithm for 2-MCIP, which implies that 2-MCIP is APX-complete. In Section 4, we will present a k-approximation algorithm for k-MCIP, which implies that k-MCIP is APX-complete, for any $k \geq 2$. □

6. CONCLUDING REMARKS

It is interesting to observe that although 2-MCIP is in some sense similar to other integer partition/summation problems such as Knapsack and Bin Packing, it is much more difficult to approximate. For example, Knapsack and Bin Packing all have an FPTAS (fully polynomial-time approximation scheme) or asymptotic PTAS, but Theorem 5.6 implies that it is unlikely for 2-MCIP to have a PTAS.

REFERENCES

ALTSCHUL, S. AND LIPMAN, D. 1989. Trees, stars, and multiple sequence alignment. *SIAM Journal on Applied Math 49,* 1, 197–209.

ANDREWS, G. 1976. *The Theory of Partitions.* Addison-Wesley.

ANDREWS, G. AND ERIKSSON, K. 2004. *The Integer Partitions.* Cambridge.

ARKIN, E. AND HASSIN, R. 1998. On local search for weighted packing problems. *Math. Oper. Res. 23*, 640–648.

AUSIELLO, G., CRESCENZI, P., GAMBOSI, G., KANN, V., MARCHETTI-SPACCAMELA, A., AND PROTASI., M. 1999. *Complexity and Approximation.* Springer.

CHEN, X. 2005. The minimum common partition problem revisited. *manuscript.*

CHEN, X., ZHENG, J., FU, Z., NAN, P., ZHONG, Y., LONARDI, S., AND JIANG, T. 2005a. The assignment of orthologous genes via genome rearrangement. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 2,* 4, 302–315.

CHEN, X., ZHENG, J., FU, Z., NAN, P., ZHONG, Y., LONARDI, S., AND JIANG, T. 2005b. Computing the assignment of orthologous genes via genome rearrangement. In *Proc. of 3rd Asia Pacific Bioinformatics Conference (APBC'05).* 363–378.

CHROBAK, M., LOLMAN, P., AND SGALL, J. 2004. The greedy algorithm for the minimum common string partition problem. *Proc. of 7th International Workshop on Approximation Algorithms for Combinatiorial Optimization Problems (APPROX)*, 84–95.

CORMEN, T., LEISERSON, C., RIVEST, R., AND STEIN, C. *Introduction to Algorithms.* The MIT Press.

FU, Z. 2004. Assignment of orthologous genes for multichromosomal genomes using genome rearrangement. *UCR CS Technical report.*

GOLDSTEIN, A., KOLMAN, P., AND ZHENG, J. 2004. Minimum common string partition problem: hardness and approximations. In *Proc. of 15th International Symposium on Algorithms and Computation (ISAAC).* LNCS 3341, 473–484.

GUSFIELD, D. 1997. *Algorithms on Strings, Tree, and Sequences: Computer Science and Computational Biology.* Cambridge University Press.

HANNENHALLI, S. AND PEVZNER, P. 1995. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Proc. 27th Ann. ACM Symp. Theory of Comput. (STOC'95).* 178–189.

HURKENS, C. AND SCHRIJVER, A. 1989. On the size of systems of sets every t of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discrete Mathematics 2*, 68–72.

KANN, V. 1991. Maximum bounded 3-dimensional matching is max snp-complete. *Information Processing Letters 37*, 27–35.

KOLMAN, P. 2005. Approximating reversal distance for strings with bounded number of duplicates in linear time. In *Proc. of 30 International Symposium on Mathematical Foundations of Computer Science (MFCS).* 580–590.

PAPADIMITRIOU, C. AND YANNAKAKIS, M. 1991. Optimization, approximation, and complexity classes. *Computer and System Sciences 43*, 425–440.

REMM, M., STORM, C., AND SONNHAMMER, E. 2001. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol. 314*, 1041–1052.

SANKOFF, D. 1989. Mechanisms of genome evolution: models and inference. *Bull. Int. Stat. Instit. 47*, 461–475.

VALINSKY, L., SCUPHAM, A., VEDOVA, G., LIU, Z., FIGUEROA, A., JAMPACHAISRI, K., YIN, B., BENT, E., MANCINI-JONES, R., PRESS, J., JIANG, T., AND BORNEMAN, J. 2004. Oligonucleotide fingerprinting of ribosomal rna genes (ofrg). In *Molecular Microbial Ecology Manual (2nd ed).* Kluwer Academic Publishers, Dordrecht, The Netherlands, 569–585.