On the Minimum Common Integer Partition Problem

Xin Chen[1], Lan Liu[2], Zheng Liu[2], and Tao Jiang[2,3]

[1] School of Physical and Mathematical Sciences, Nanyang Tech. Univ., Singapore
ChenXin@ntu.edu.sg
[2] Department of Computer Science, Univ. of California at Riverside, USA
lliu, zliu, jiang@cs.ucr.edu
[3] Currently visiting at Tsinghua University, Beijing, China

Abstract. 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 $S_1, \cdots, S_k$ 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 $S_1, \cdots, S_k$ with the minimum cardinality. It is easy to see that the MCIP problem is NP-hard since it generalizes the well-known Set Partition 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 $k \geq 3$.

1 Introduction

Computational molecular biology has emerged as one of the most exciting interdisciplinary 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 [12], sorting by reversals [20], and recently the minimum common partition problem [10]. These problems have attracted considerable attention from computer scientists who took the challenge to design efficient and effective algorithms for solving them [5, 14, 13]. 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 $\{n_1, n_2, \cdots, n_r\}$ 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 n [2, 4]. Given a multiset $S = \{x_1, x_2, \cdots, x_m\}$ of integers with a partition 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

T. Calamoneri, I. Finocchi, G.F. Italiano (Eds.): CIAC 2006, LNCS 3998, pp. 236–247, 2006.

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 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 is a common pratice and should not cause any confusion given the context. For simplicity, we also denote by 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 [2]. 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.*, MCIP$(S,T) = \{2,2,3,3\}$. Note that the minimum common integer partition is not necessarily unique. So, the notation 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 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 Set Partition problem [7]. In this paper, we show that the MCIP problem is APX-hard and hence has no polynomial-time approximation algorithm (PTAS) unless P = NP. We also present a $\frac{5}{4}$-approximation algorithm for the 2-MCIP 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 $k \geq 3$.

1.1 Biological Background

Although the MCIP problem is quite a natural extension of the Set Partition 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 assign-

ment) of orthologs is a common task in computational biology. While it is usually done using sequence homology search [19], we have recently proposed an alternative and promising approach to assign orthologs via genome rearrangement [9, 10]. 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 [6, 13, 11, 16]. 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 [8], when $l \geq 1$. In other words, even when there is only one symbol with multiple copies in input strings, 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 [8]. 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 [21], 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 nonbinding 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 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. Due to page constraint, we omit the proofs of all the lemmas and Theorem 4 (See [22] for the details of the proofs).

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 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.*

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.

Algorithm 2-APPROX-MCIP(S, T)
input Two related multisets S and T
output A common integer partition CIP of S and T
begin
  $CIP := \emptyset$;
  while $S \neq \emptyset$ **do**
    arbitrarily pick $x_i \in S$ and $y_j \in T$;
    $S := S \setminus \{x_i\}$; $T := T \setminus \{y_j\}$;
    $z := \min(x_i, y_j)$; $CIP := CIP \biguplus \{z\}$;
    if $x_i \neq z$ $S := S \biguplus \{x_i - z\}$;
    if $y_i \neq z$ $T := T \biguplus \{y_i - z\}$;
end.

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

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);
  $CIP := CIP(S_1, T_1) \biguplus CIP(S_2, T_2)$;
  $CIP := CIP \biguplus$ 2-APPROX-MCIP(S_3, T_3);
  return CIP;
end.

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

Lemma 2. $|MCIP(S,T)| \leq |S| + |T| - 1$.

As its name suggests, 2-APPROX-MCIP(S,T) is a 2-approximation algorithm for the problem of 2-MCIP, which is implied by Lemma 1 and Lemma 2.

Lemma 3. *The algorithm* 2-APPROX-MCIP*(S,T) achieves an approximation ratio of 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 x_i as well as a part 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.*, $a_1, a_h \in S \biguplus T$) 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 $CIP(S,T)$.

Lemma 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 parts of 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 $|CIP(S_1,T_1)| \geq |S_1| + |T_1| - 1$.*

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.* $S = \biguplus_{i=1}^{l} S_i$. 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 $|2\text{-}MRMP|$) the size of the partition, *i.e.*, the number of related multiset pairs in the partition.

Lemma 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 $|RMP(S, T)| \geq |S| + |T| - |CIP(S, T)|$.*

The following lemma establishes the relationship between MCIP and MRMP, showing their (complementary) equivalence.

Lemma 6. *If S and T are related multisets, then $|MCIP(S, T)|+|MRMP(S, T)| = |S| + |T|$.*

Since a pair of basic related multisets S and T cannot be partitioned further into related submultisets, *i.e.*, $|MRMP(S, T)| = 1$, the following lemma is trivially implied by Lemma 6.

Lemma 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 8. *If the minimum size of any related submultiset of S and T is c, then $|MCIP(S, T)| \geq \frac{c-1}{c}(|S| + |T|)$.*

Lemma 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 identical, then* $\{x_i\} \biguplus MCIP(S \backslash \{x_i\}, T \backslash \{y_j\})$ *is a minimum common integer partition of* S *and* T, i.e., $|MCIP(S,T)| = |MCIP(S \setminus \{x_i\}, T \setminus \{y_j\})| + 1$.

Unfortunately, the result in Lemma 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 9.

3 Hardness of Approximation

It is easy to see that MCIP is NP-hard because there is a straightforward reduction from the *Set Partition* 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 [17], 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:

- A multiset $\tilde{X} = \{\tilde{x}_i | \tilde{x}_i = 4^i, \forall x_i \in X\}$;
- A multiset $\tilde{Y} = \{\tilde{y}_i | \tilde{y}_i = 4^{|X|+i}, \forall y_i \in Y\}$;
- A multiset $\tilde{Z} = \{\tilde{z}_i | \tilde{z}_i = 4^{|X|+|Y|+i}, \forall z_i \in Z\}$;
- A multiset $\tilde{D} = \{\tilde{d}_i | \tilde{d}_i = \tilde{x}_{i^X} + \tilde{y}_{i^Y} + \tilde{z}_{i^Z}, \forall d_i \in D\}$;
- An integer $e = \sum_{i=1}^{|D|} \tilde{d}_i - \sum_{i=1}^{|X|} \tilde{x}_i - \sum_{i=1}^{|Y|} \tilde{y}_i - \sum_{i=1}^{|Z|} \tilde{z}_i$.
- 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 10. *For any instance of MAX 3DM-3,* $|2\text{-}MCIP| \leq 70 \cdot |MAX\ 3DM\text{-}3|$.

Given a common integer partition 2-CIP of $\langle S, T\rangle$, we define a function g to construct a subset (denoted as $3DM$-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-CIP.

Lemma 11. *For any instance D of MAX 3DM-3, the subset* $3DM$-3 *constructed by the function g is a matching of D.*

Let $|2\text{-}MRMP|$ be the size of the maximum related multiset partition of S and T. 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 12. $|2\text{-}MRMP| = |MAX\ 3DM\text{-}3| + 1$.

Lemma 13. $|MAX\ 3DM\text{-}3| - |3DM\text{-}3| \leq |2\text{-}CIP| - |2\text{-}MCIP|$.

Lemma 14. *MAX 3DM-3* $\leq_L$ *2-MCIP.*

Theorem 1. *The k-MCIP problem is APX-complete, for any $k \geq 2$.*

Proof. Since the MAX 3DM-3 problem is APX-complete [17] and MAX 3DM-3 $\leq_L$ 2-MCIP by Lemma 14, 2-MCIP is APX-hard. In addition, by Lemma 3, there exists a polynomial-time 2-approximation algorithm for 2-MCIP, which implies that 2-MCIP is APX-complete. In Section 5, we will present a k-approximation algorithm for k-MCIP, which implies that k-MCIP is APX-complete, for any $k \geq 2$. □

4 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 basic related submultisets of sizes three and four between S and T. As mentioned earlier, 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 [1], in which the collection C consists of all the basic related submultisets of sizes three and four between S and T. Since the cardinality of each 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 [15] 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 [3] 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 basic related multiset of size three is 2 and the weight for a multiset of size four is 1, and the goal is to find a collection of disjoint multisets of 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. Due to the space limitation, the running time analysis is omitted here, which can be found in [22] .

Let q_3 and q_4 denote the numbers 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 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. [1]

Lemma 15. $2q_3^* + q_4^* \leq 4(q_3 + q_4)$.

Let q_3' and q_4' be the numbers of basic related submultisets of sizes three and four in the related multiset partition induced by a given minimum common partition 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 16. $|MCIP(S,T)| \geq \frac{4}{5}(m + n) - \frac{1}{5}(2q_3^* + q_4^*)$, *where* $m = |S|$ *and* $n = |T|$.

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

Lemma 17. $|MCIP(S,T)| \leq m + n - q_3 - q_4 - 1$.

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

[1] The $(k/2 + \epsilon)$-approximation algorithm given by Hurkens and Schrijver [15] can also find a packing of C satisfying the inequality in Lemma 15, but only in quasi-polynomial time.

Algorithm k-APPROX-MCIP($S_1, \cdots, S_k$)
input Related multisets $S_1, \cdots, S_k$
output A common integer partition CIP of $S_1, \cdots, S_k$
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.

Algorithm
$\frac{3k(k-1)}{3k-2}$-APPROX-MCIP($S_1, \cdots, S_k$)
input Related multisets $S_1, \cdots, S_k$
output A common integer partition CIP of $S_1, \cdots, S_k$
begin
  remove_common_integer($S_1, \cdots, S_k$);
  $CIP :=$ k-APPROX-MCIP($S_1, \cdots, S_k$);
  return CIP;
end.

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

$\Theta((m+n)^4)$ 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 2. *The algorithm $\frac{5}{4}$-APPROX-MCIP is a $\frac{5}{4}$-approximation algorithm for 2-MCIP.*

Proof. By Lemmas 16 and 17, 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 15, we know that $2q_3^*+q_4^* \leq 4(q_3+q_4)$. Therefore, $\alpha \leq \frac{5}{4}$. □

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

Theorem 3. *The algorithm k-APPROX-MCIP is a k-approximation algorithm for the k-MCIP ($k \geq 2$) problem.*

Proof. By Lemma 1 and Lemma 18, 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 19. *If there is no integer element common to all the multisets in* $\{S_1, S_2, \cdots, S_k\}$*, then it holds that* $|MCIP(S_1, S_2, \cdots, S_k)| \geq \frac{3k-2}{3k(k-1)} \sum_{i=1}^{k} |S_i|$.

Theorem 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* ($k \geq 2$) *problem.*

Clearly, the algorithm $\frac{3k(k-1)}{3k-2}$-Approx-MCIP$(S_1, \cdots, S_k)$ runs in $O(\sum_i |S_i| \cdot log(\sum_i |S_i|))$ time. Let us compare Theorem 4 with Theorem 3. 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 3. 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.

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 1 implies that it is unlikely for 2-MCIP to have a PTAS.

Acknowledgments

We would like to thank David P. Woodruff for several useful discussions. This project is supported in part by NSF grants CCR-0309902 and DBI-0133265, NSFC grant 60528001, National Key Project for Basic Research (973) grant 2002CB512801, and a fellowship from the Center for Advanced Study, Tsinghua University.

References

1. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation*, Springer, 1999.
2. G.E. Andrews. *The Theory of Partitions*, Addison-Wesley, 1976.
3. E.M. Arkin and R. Hassin. On local search for weighted packing problems. *Math. Oper. Res.* 23, pp. 640-648, 1998.
4. G.E. Andrews and K. Eriksson. *The Integer Partitions*, Cambridge, 2004.
5. S. Altschul and D. Lipman. Trees, stars, and multiple sequence alignment. *SIAM Journal on Applied Math.* 49(1), pp. 197-209, 1989.

6. M. Chrobak, P. Lolman, and J. Sgall. The greedy algorithm for the minimum common string partition problem. *Proc. of 7th International Workshop on Approximation Algorithms for Combinatiorial Optimization Problems (APPROX)*, pp. 84-95, 2004.
7. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. *Introduction to algorithms*, The MIT Press, 2nd edition, p. 1017, 2001.
8. X. Chen. The minimum common partition problem revisited. *manuscript*, 2005.
9. X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong, S. Lonardi, and T. Jiang. Computing the assignment of orthologous genes via genome rearrangement. *Proc. of 3rd Asia Pacific Bioinformatics Conference (APBC'05)*, pp. 363-378, 2005.
10. X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong, S. Lonardi, and T. Jiang. The assignment of orthologous genes via genome rearrangement. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4), pp. 302-315, 2005.
11. Z. Fu. Assignment of orthologous genes for multichromosomal genomes using genome rearrangement. *UCR CS Technical report*, 2004.
12. D. Gusfield. *Algorithms on Strings, Tree, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, 1997.
13. A. Goldstein, P. Kolman, and J. Zheng. Minimum common string partition problem: hardness and approximations. *Proc. of 15th International Symposium on Algorithms and Computation* (ISAAC), LNCS 3341, pp. 473-484, 2004.
14. S. Hannenhalli and P.A. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). *Proc. 27th Ann. ACM Symp. Theory of Comput.* (STOC'95), pp. 178-189, 1995.
15. C. Hurkens and A. Schrijver. 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, pp. 68-72, 1989.
16. P. Kolman. Approximating reversal distance for strings with bounded number of duplicates in linear time. *Proc. of 30 International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pp. 580-590, 2005.
17. V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37: 27-35, 1991.
18. C.H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Computer and System Sciences*, 43: 425-440, 1991.
19. M. Remm, C. Storm, and E. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.*, 314, pp. 1041-1052, 2001.
20. D. Sankoff. Mechanisms of genome evolution: models and inference. *Bull. Int. Stat. Instit.* 47, pp. 461-475, 1989.
21. L. Valinsky, A. Scupham, G.D. Vedova, Z. Liu, A. Figueroa, K. Jampachaisri, B. Yin, E. Bent, R. Mancini-Jones, J. Press, T. Jiang, and J. Borneman. Oligonucleotide Fingerprinting of Ribosomal RNA Genes (OFRG), pp. 569-585. In G. A. Kowalchuk, F. J. de Bruijn, I. M. Head, A. D. L. Akkermans, J. D.van Elsas (eds.) *Molecular Microbial Ecology Manual* (2nd ed). Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.
22. Available at http://www.cs.ucr.edu/∼lliu/paper/mcip_ciac_full.pdf.