Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 369 (2006) 456–462
Theoretical Computer Science
www.elsevier.com/locate/tcs

Note

A network flow approach to the Minimum Common Integer Partition Problem

Wenbo Zhao[a, b, *, 1], Peng Zhang[a, b, 1], Tao Jiang[c, 2, 3]

[a] *Institute of Software, Chinese Academy of Sciences, P.O. Box 8718, Beijing 100080, China*
[b] *Graduate University of Chinese Academy of Sciences, Beijing, China*
[c] *Department of Computer Science and Engineering, University of California at Riverside, USA*

Received 22 May 2006; received in revised form 4 September 2006; accepted 5 September 2006

Communicated by Ding-Zhu Du

Abstract

In the k-Minimum Common Integer Partition Problem, abbreviated as k-MCIP, we are given k multisets $X_1, \ldots, X_k$ of positive integers, the goal is to find an integer multiset T of the minimum size such that for every i, we can partition each of the integers in X_i so that the disjoint (multiset) union of their partitions equals T. This problem has applications in computational molecular biology, in particular, ortholog assignment and DNA hybridization fingerprint assembly. The problem is known to be NP-hard for any $k \geqslant 2$. In this article, we improve the approximation ratio for k-MCIP by viewing this problem as a flow decomposition problem in some flow network. We show an efficient $0.5625k$-approximation algorithm, improving upon the previously best known $0.6139k$-approximation algorithm for this problem.

Keywords: Minimum Common Integer Partition; Approximation algorithm; Network flow

1. Introduction

Recently, Chen et al. [4] introduced a new combinatorial optimization problem, called the *k-Minimum Common Integer Partition Problem*, abbreviated as k-MCIP, which was inspired by their recent work on ortholog assignment and DNA fingerprint assembly [2,3,6,9]. Formally, the problem is as follows. Consider two multisets $X = \{x_1, \ldots, x_m\}$ and T of positive integers. If there is a partition of T into multisets T_i such that for each i the sum of integers in T_i equals x_i, then T is called an *integer partition* of X. We say that T is a *common integer partition* of multisets $X_1, \ldots, X_k$ if it is an integer partition of each X_i and the integers in each X_i have the same sum. The k-MCIP($X_1, \ldots, X_k$) is to find a common integer partition T of the minimum cardinality.

* Corresponding author. Institute of Software, Chinese Academy of Sciences, P.O. Box 8718, Beijing 100080, China. Tel.: +86 010 82625471.
E-mail addresses: zwenbo@gcl.iscas.ac.cn (W. Zhao), zhp@gcl.iscas.ac.cn (P. Zhang), jiang@cs.ucr.edu (T. Jiang).
[1] Supported by NSFC Grants no. 60325206 and no. 60310213.
[2] Supported by NSF Grant CCR-0309902, NIH Grant LM008991-01, NSFC Grant 60528001, and a fellowship from the Center for Advanced Study, Tsinghua University.
[3] Currently visiting at Tsinghua University, Beijing, China.

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.09.001

As an example, for a triple of multisets $X_1 = \{1, 5, 10\}$, $X_2 = \{3, 3, 10\}$, and $X_3 = \{2, 2, 12\}$, it is easy to verify that the integer partition $T = \{1, 1, 2, 2, 10\}$ is a minimum common integer partition of the X_i. In [4], it is shown that k-MCIP is NP-hard since it generalizes the well-known set partition problem [5], and in fact APX-hard [8] for every $k \geqslant 2$. Hence, the k-MCIP problem has no polynomial-time approximation algorithm (PTAS) unless P = NP. The authors in [4] also presented a $\frac{5}{4}$-approximation algorithm for 2-MCIP by using a heuristic for the *Maximum Set Packing Problem*, and a $3k(k-1)/(3k-2)$-approximation algorithm for the general k-MCIP problem, where $k \geqslant 3$. Note that $3k(k-1)/(3k-2) \simeq k - \frac{1}{3}$. More recently, Woodruff in [10] proved a better approximation ratio for k-MCIP by looking at what he called *redundancy* of $X_1, \ldots, X_k$, which is a quantity capturing the frequency of integers across different X_i's, and he obtained a $0.6139k$-approximation ratio.

Given an instance S of k-MCIP, we notice that the k-MCIP problem can be formalized as a *flow decomposition problem* in an acyclic k-layer network $N = \{s, t, V, A, c\}$ corresponding to S. The object is to find a minimum number of directed simple paths from s to t, each of which can carry a positive integral flow from s to t such that the total amount of flow carried by these paths is equal to the value of a max-flow in this network. Such a set of paths is referred to as *good*. Notice that the well-known Flow Decomposition Theorem indicates that we can always find a good set of paths by the Flow Decomposition Algorithm in [1]. In the general case, the size of the good set of paths returned by the Flow Decomposition Algorithm in [1] can be bounded by the number of arcs in the network. Thus, in order to achieve an improved approximation ratio, we have to reduce the number of arcs without breaking the fundamental requirement of this network, i.e., the resultant network after arc reduction can still induce a common integer partition for S. More details about the connection between k-MCIP and the flow network will be described in the next section.

In this paper, we present a randomized approximation algorithm based on the connection mentioned above and show that it achieves an approximation ratio of $0.5625k$. This improves on the previous results in [4,10]. Furthermore we will show that this randomized algorithm can be derandomized. Particularly, for the special case where each multiset X_i intersects at most $q-1$ other multisets, we improve the analysis of our algorithm to get a ratio of $(k+q)/2$.

Our terminology and notations for randomized algorithm are standard which can be found in [7]. The basic notations about network flow follow those in [1].

2. Preliminaries

Given an instance S of the *k-MCIP* consisting of k multisets of integers $S = \{X_1, \ldots, X_k\}$, where $X_1 = \{x_{11}, \ldots, x_{1|X_1|}\}$, …, $X_k = \{x_{k1}, \ldots, x_{k|X_k|}\}$, a k-layer network $N = (s, t, V, A, c)$ can be constructed as follows. There are $k+1$ nodes in the k-layer network, i.e., $s = v_1, v_2, \ldots, v_k, t = v_{k+1}$. For $i = 1, \ldots, k$ and $j = 1, \ldots, |X_i|$, there are $|X_i|$ arcs, namely a_{ij}, from node v_i to node v_{i+1}, each with capacity $c(a_{ij}) = x_{ij}$, respectively. In the following, we say that an arc is in layer i if the arc emanates from v_i and terminates at v_{i+1}.

In the above network N, let $\mathcal{P}$ denote the set of paths from s to t. Consider the Max-Flow Problem in network N and its LP relaxation as follows:

$$\max \sum_{P \in \mathcal{P}} f(P) \text{ s.t. } \sum_{P: a \in \mathcal{P}} f(P) \leqslant c(a), \quad a \in A,\ f(P) \geqslant 0, \quad P \in \mathcal{P}, \tag{1}$$

where the variable $f(P)$ is the amount of flow sent on P. Denote the value of an optimal solution of LP above by f^* and we say that a subset $\mathcal{P}' \subseteq \mathcal{P}$ is *good* if $f(P)$ is integral for every $P \in \mathcal{P}'$ and $\sum_{P: a \in P} f(P) = c(a)$ for each arc a. Note that the second condition implies that $\sum_{P \in \mathcal{P}'} f(P) = f^*$. Define $opt(S)$ to be the size of a minimum common partition of S. When S is clear from the context, we will often just write opt. The key observation is that

$$\min_{\mathcal{P}' \text{ is good}} |\mathcal{P}'| = opt(S). \tag{2}$$

Thus, the k-MCIP is equivalent to finding a good set $\mathcal{P}'$ with the minimum size.

As an example, for a triple of multisets $X_1 = \{1, 5, 10\}$, $X_2 = \{3, 3, 10\}$, and $X_3 = \{2, 2, 12\}$, the integer partition $T = \{1, 1, 2, 2, 10\}$ is a minimum common integer partition of the X_i. In the network N induced by these three multisets, there is a good set $\mathcal{P}'$ consisting of five simple directed paths from s to t and the multiset $\{f(P) | P \in \mathcal{P}'\}$ corresponding to $\mathcal{P}'$ is the same as T. See Figs. 1(a) and (b).

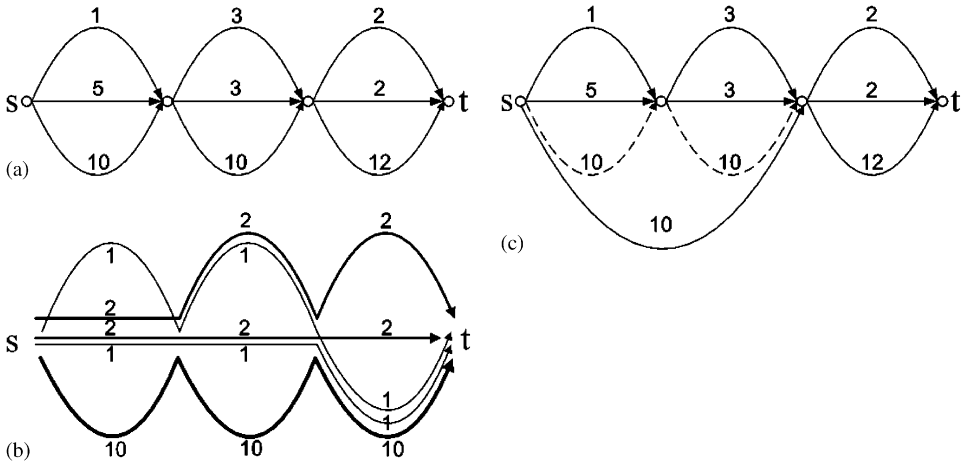

Fig. 1. (a) A 3-layer network induced by $X_1 = \{1, 5, 10\}$, $X_2 = \{3, 3, 10\}$, and $X_3 = \{2, 2, 12\}$. (b) A flow decomposition. (c) Two arcs (the dashed lines) with capacity 10 merge into one arc (the bold line).

3. The randomized algorithm

Since the k-layer network $N = (s, t, V, A, c)$ induced by $X_1, \ldots, X_k$ is directed and acyclic, by the well-known Flow Decomposition Theorem, the Flow Decomposition Algorithm in [1] can always return a good $\mathcal{P}'$ w.r.t. N with at most $m - l + 1$ arcs, where $m = |A|$ and l is the length of the shortest path from s to t.

For a node $v_i \in V, 2 \leqslant i \leqslant k$, if there are two arcs $a, a' \in A$ where arc a is from v_{i-1} to v_i and arc a' is from v_i to v_{i+1} with the same capacity, i.e., $c(a) = c(a')$, we can concatenate arcs a and a' into a single arc with capacity $c(a)$, i.e., delete arcs a, a' and add a new arc with capacity $c(a)$ from v_{i-1} to v_{i+1}. See Fig. 1(c). We denote this operation by *Merge(a, a′)*.

Let $N_1 = (s_1, t_1, V_1, A_1, c_1)$ be the network induced by applying some *Merge* operations on N. It is easy to verify that any set $\{f(P)|P \in \mathcal{P}'_1\}$ corresponding to a good path set $\mathcal{P}_1$ for network N_1 is also a feasible solution for the instance of the original k-MCIP and $|A_1| < |A|$. Hence, the operation Merge can help us remove "redundant" arcs. This suggests the following greedy algorithm.

Algorithm CIP $(X_1, \ldots, X_k)$

(1) Choose a permutation π uniformly at random.
(2) Construct the k-layer Network $N = (s, t, V, A, c)$ according to $(X_{\pi(1)}, \ldots, X_{\pi(k)})$.
(3) For each node $v_i, 2 \leqslant i \leqslant k$, merge all pairs of arcs (a, a') with the same capacity, where a terminates at v_i and a' emanates from v_i. Denote the resulting network by $N_1 = (s, t, V, A_1, c_1)$.
(4) Use the Flow Decomposition Algorithm to get a good set of paths $\mathcal{P}'_1$ and the corresponding multiset $T = \{f(P)|P \in \mathcal{P}'_1\}$.
(5) Output T.

In the next section we will develop a key lemma for lower-bounding the expectation of the number of pairs merged in step 3, and we use it to analyze the performance of the algorithm **CIP**.

4. Analysis of the algorithm CIP

Consider an instance S of k-MCIP consisting of k multisets of integers $S = \{X_1, \ldots, X_k\}$ and the k-layer network $N = (s, t, V, A, c)$ induced by the instance S. Recall that the capacity of arc $a_{ij} \in A, 1 \leqslant i \leqslant k, 1 \leqslant j \leqslant |X_i|$, is $c(a_{ij}) = x_{ij}$. To analyze the algorithm, we need some notations which are adapted from those in [10].

Let $E \subseteq A$ be a subset of arcs in N. We say that E is *lonely* if $c(a_{i_1 j_1}) = c(a_{i_2 j_2}), a_{i_1 j_1}, a_{i_2 j_2} \in E$ and for any pair $a_{i_1 j_1}, a_{i_2 j_2} \in E, i_1 \neq i_2$, i.e., no two arcs belong to the same layer in E. In this case we use the notation $c(E)$ to denote

the common capacity of arcs a_{ij}, $a_{ij} \in E$. For two disjoint lonely sets E_1, E_2 such that $c(E_1) = c(E_2)$, we say that E_1 *covers* E_2 if for any arc $a_{ij_2} \in E_2$ there must be an arc $a_{ij_1} \in E_1$ in the same layer as that of a_{ij_2}. Moreover, we call a collection $\mathcal{C}$ of lonely sets *consistent* if there are no two distinct sets $E_1, E_2 \in \mathcal{C}$ that intersect. We define the *weighted-size* of a consistent collection $\mathcal{C}$ of lonely sets E_j to be $\sum_{j=1}^{|\mathcal{C}|} |E_j|$.

Definition 1. The *r-redundancy* of network denoted as N corresponding to S, denoted by $\mathbf{Red}(\mathbf{r}, \mathbf{N})$, is the maximum, over all consistent $\mathcal{C}$ consisting of at most r lonely sets, of the weighted-size of $\mathcal{C}$.

In fact, for any fixed r, we can find a consistent collection $\mathcal{C}_r$ of lonely sets $E_1, \ldots, E_r$ such that $\mathbf{Red}(\mathbf{r}, \mathbf{N}) = \sum_{j=1}^{r} |E_j|$. We now describe an iterative procedure to find the consistent collection $\mathcal{C}_r$. To accomplish this, at first, we pick a set $E_1 \subseteq A$ such that E_1 is lonely and $|E_1|$ is the maximum. In fact, E_1 can be found by a greedy fashion (i.e., for any $x_{i_1 j_1} \in X_{i_1}$, $1 \leqslant i_1 \leqslant k$, let $E_1 = \{x_{i_1 j_1}\}$ initially and add other x_{ij} into E_1 as many as possible while keeping E_1 lonely). Then pick another lonely set $E_2 \subseteq A - E_1$ such that $|E_2|$ (or $|E_1| + |E_2|$) is maximized (E_2 can also be found by the same greedy fashion). Obviously, the collection of E_1 and E_2 is consistent. Repeating the above process we will have a consistent collection $\mathcal{C}$ of lonely sets $E_1, E_2, \ldots, E_r$. Let $\mathcal{C}_r = \{E_1, \ldots, E_r\}$. We can prove the following lemma.

Lemma 2. $\mathbf{Red}(\mathbf{r}, \mathbf{N}) = \sum_{j=1}^{r} |E_j|$.

Proof. Suppose $\mathcal{C}'_r$ is a consistent collection of lonely sets $E'_1, \ldots, E'_r$ such that $\mathbf{Red}(\mathbf{r}, \mathbf{N}) = \sum_{j=1}^{r} |E'_j|$. Let $E'_{j_1}, \ldots, E'_{j_l}$ be the sets such that $c(E'_{j_1}) = \cdots = c(E'_{j_l}) = x$ and there is no other set $E'_j \in \mathcal{C}'_r$ such that $c(E'_j) = x$. W.l.o.g., we may assume $|E'_1| \geqslant \cdots \geqslant |E'_r|$.

By using the same greedy algorithm as described above for constructing $\mathcal{C}_r$, we can divide the set of arcs in A with the same capacity x into l^* disjoint lonely sets, say, $E''_1, \ldots, E''_{l^*}$. Also, we may assume that $|E''_1| \geqslant \cdots \geqslant |E''_{l^*}|$ and E''_i cover E''_j for $1 \leqslant i < j \leqslant l^*$. By the definition of $\mathbf{Red}(\mathbf{r}, \mathbf{N})$ we have $l^* \geqslant l$ (otherwise we could make $\sum_{j=1}^{r} |E'_j|$ larger). We prove that $\sum_{i=1}^{l} |E'_{j_i}| = \sum_{i=1}^{l} |E''_{j_i}|$. First, observe that if for some $1 \leqslant l_1 < l_2 \leqslant l$, $|E'_{j_{l_1}}| \geqslant |E'_{j_{l_2}}|$ but $E'_{j_{l_1}}$ does not cover $E'_{j_{l_2}}$, then there must be an arc belonging to $E'_{j_{l_2}}$ in some layer but $E'_{j_{l_1}}$ has no arc in the same layer, and thus we can remove all such arcs from $E'_{j_{l_2}}$ and add them to $E'_{j_{l_1}}$ without changing $\sum_{j=1}^{r} |E'_j|$. Hence, we can assume that $E'_{j_{l_1}}$ cover $E'_{j_{l_2}}$ for all $1 \leqslant l_1 < l_2 \leqslant l \leqslant l^*$. It is easy to verify that the largest sum of the sizes of l such sets is just $\sum_{i=1}^{l} |E''_{j_i}|$. Therefore, the claim holds, which gives rise to the conclusion of the lemma. □

Recall Eq. (2) and that $m = \sum_{i=1}^{k} |X_k|$. For our purpose, we need a lower bound on *opt* in terms of the redundancy of S, as was done by Woodruff in [10]. For the sake of completeness, we include a proof of the lower bound here, which is slightly different from that in [10].

Lemma 3. $opt \geqslant (2m - \mathbf{Red}(\mathbf{opt}, \mathbf{N}))/k$.

Proof. Given the network $N = (s, t, V, A, c)$ defined for the instance S, let $\mathcal{P}'$ be a good set consisting of paths from s to t such that $|\mathcal{P}'| = opt(S)$. Define sets of arcs $A^{(1)}$, $A^{(1)}_P$, and $A^{(2)}$ as follows:

$$A^{(1)} = \{a \in A : \text{only one path } P \in \mathcal{P}' \text{ passing through arc } a\};$$
$$A^{(1)}_P = \{a \in A^{(1)} : a \in P\} \text{ for each } P \in \mathcal{P}';$$
$$A^{(2)} = A - A^{(1)}.$$

Since, for each arc in $A^{(2)}$, there are at least two paths in $\mathcal{P}'$ passing through it and the length of each path $P \in \mathcal{P}'$ is k. It is clear that $k \cdot |\mathcal{P}'| \geqslant |A^{(1)}| + 2 \cdot |A^{(2)}| = 2m - |A^{(1)}|$ (See Fig. 1(b)). Moreover, let $\mathcal{C}'$ be the collection of sets $A^{(1)}_P$, $P \in \mathcal{P}'$. It is easy to verify that $A^{(1)}_P$ is lonely for each $P \in \mathcal{P}'$ and $\mathcal{C}'$ is a consistent collection containing at most *opt* many sets. By the definition of $\mathbf{Red}(\mathbf{opt}, \mathbf{N})$, $|A^{(1)}| = \sum_{P \in \mathcal{P}'} |A^{(1)}_P| \leqslant \mathbf{Red}(\mathbf{opt}, \mathbf{N})$. Thus, $2m - \mathbf{Red}(\mathbf{opt}, \mathbf{N}) \leqslant k \cdot |\mathcal{P}'| = k \cdot opt$, and rearranging the terms gives rise to the lemma. □

We immediately have the following corollary.

Corollary 4. *If for all $j \neq j'$, X_j and $X_{j'}$ are disjoint, then the algorithm* **CIP** *achieves $(k+1)/2$-approximation. Moreover, if for each X_i, there are at most $q-1$ sets X_j, $j \neq i$, that intersect with X_i, then algorithm* **CIP** *achieves $(k+q)/2$-approximation.*

Proof. We will use the same definitions and notations as in the proof of Lemma 3. If for each X_i, at most $q-1$ X_j, $j \neq i$, intersect with X_i, then for any path $P \in \mathcal{P}'$, there are at most q arcs in path P that are only passed through by P. Hence, we have $|A^{(1)}| \leqslant q \cdot opt$. Recall that $k \cdot opt \geqslant |A^{(1)}| + 2 \cdot |A^{(2)}| = 2m - |A^{(1)}|$. It follows that $k \cdot opt \geqslant 2m - q \cdot opt$. Rearranging terms proves the corollary. □

We now turn to the performance of the algorithm **CIP** and prove our main result, Theorem 5, given below. In the following, we will use notation o() and $\omega()$ to denote functions of k that are independent of m. For instance, o(1) stands for a function that tends to 0 as k goes to infinity.

Theorem 5. *The expectation of the output of* $\mathbf{CIP}(X_1, \ldots, X_k)$ *is a* $0.5625k(1 + \mathrm{o}(1))$*-approximation.*

Proof. Denote the number of pairs (a, a') merged in the ith iteration of step 3 by $M(X_{\pi(i-1)}, X_{\pi(i)})$. By symmetry, it is clear that the expectation of the number of pairs merged in each iteration of step 3 is the same. Thus, we have $E[M(X_{\pi(i-1)}, X_{\pi(i)})] = E[M(X_{\pi(j-1)}, X_{\pi(j)})]$, $2 \leqslant i, j \leqslant k$. By the linearity of expectation, the total number of pairs merged in step 3 is $(k-1)E[M(X_{\pi(1)}, X_{\pi(2)})]$.

In order to lower bound $E[M(X_{\pi(1)}, X_{\pi(2)})]$, consider the largest (in terms of weighted-size) consistent collection of *opt* lonely sets $E_1, \ldots, E_{opt}$. Suppose that the sizes of these sets are $f_1, \ldots, f_{opt}$, respectively. Let $\bar{f} = \lfloor \sum_{j=1}^{opt} f_j / opt \rfloor$. Observe that $\bar{f}$ is a positive integer. Now from Lemma 3 we know that

$$opt \geqslant (2m - \mathbf{Red}(\mathbf{opt}, \mathbf{N}))/k.$$

But $\mathbf{Red}(\mathbf{opt}, \mathbf{N})) = \sum_{j=1}^{opt} f_j \leqslant (\bar{f}+1)opt$, and by rearranging the terms, we have

$$opt \geqslant \frac{2m}{k + \bar{f} + 1}.$$

Let $t = \bar{f}/k$. Clearly, $t \leqslant 1$. We may assume that $t = \omega(1)$, since otherwise $t = \mathrm{o}(1)$ and the above bound would already imply an asymptotic approximation ratio of $0.5k$. We may bound $E[M(X_{\pi(1)}, X_{\pi(2)})]$ as follows [10].

$$E[M(X_{\pi(1)}, X_{\pi(2)})] \geqslant \sum_j \frac{\binom{f_j}{2}}{\binom{k}{2}} = \sum_j \frac{f_j(f_j-1)}{k(k-1)}. \tag{3}$$

It is easy to verify that the above expression is minimized when all of the numbers f_j are equal to their average $\sum_{j=1}^{opt} f_j / opt$ [10] (note, in this minimization process we are relaxing the integrality constraints on the numbers f_j, which can only make the above expression smaller than its realistic values). Thus we have

$$\begin{aligned} E[M(X_{\pi(1)}, X_{\pi(2)})] &\geqslant \sum_{j=1}^{opt} \frac{\bar{f}(\bar{f}-1)}{k(k-1)} \\ &\geqslant \sum_{j=1}^{opt} \frac{(\bar{f}-1)^2}{(k-1)^2} \\ &= \sum_{j=1}^{opt} \frac{(tk-1)^2}{(k-1)^2} \\ &= t^2\, opt(1 - \mathrm{o}(1)). \end{aligned} \tag{4}$$

Recall that T is the output of $\mathbf{CIP}(X_1, \ldots, X_k)$ and $m = \sum_{i=1}^{k} |X_i|$. Since the expected number of arcs remaining in N_1, i.e., $|A_1|$ is at most $m - (k-1)E[M(X_{\pi(1)}, X_{\pi(2)})]$ at the end of step 3 and the size of good set of paths $\mathcal{P}'$ in step 4 is

less than $|A_1|$ by the Flow Decomposition Theorem [1], it follows that $E[|T|] \leqslant E[|A_1|] \leqslant m - (k-1)t^2 opt(1 - o(1))$. Now, recall that

$$\frac{k + \bar{f} + 1}{2} \geqslant \frac{m}{opt}.$$

We thus have

$$\begin{aligned}\frac{E[|T|]}{opt} &\leqslant \frac{m}{opt} - (k-1)t^2(1 - o(1)) \\ &\leqslant \frac{k + \bar{f} + 1}{2} - (k-1)t^2(1 - o(1)) \\ &\leqslant \frac{k}{2} + (\frac{t}{2} - t^2)k + 1 + (k-1)o(1)t^2 \\ &\leqslant \frac{k}{2} + \max_t(\frac{t}{2} - t^2)k + 1 + (k-1)o(1).\end{aligned}$$

Since $t \leqslant 1$,

$$\max_{0 \leqslant t \leqslant 1} \left(\frac{t}{2} - t^2\right) k = \frac{1}{16}k = 0.0625k.$$

Thus, we have an asymptotic randomized $0.5625k$-approximation algorithm for k-MCIP. Theorem 5 follows. □

5. Derandomizing via the method of conditional expectation

The randomized algorithm **CIP** can be derandomized via the standard method of conditional expectation. For a permutation π chosen uniformly at random, we denote the number of total pairs of arcs merged in step 3 by M_π. It is clear that $M_\pi = \sum_{i=2}^{k} M(X_{\pi(i-1)}, X_{\pi(i)})$ and

$$E[M_\pi] = \sum_{i=1}^{k} E[M_\pi | \pi(1) = i] \cdot \Pr[\pi(1) = i], \tag{5}$$

where $\Pr[\pi(1) = i] = \frac{1}{k}$ by symmetry again. Denote by M_{ij} the number of pairs of arcs that can be merged when X_i and X_j are put at adjacent layers of the network N in any permutation π. For convenience, let $M_{ij} = M_{ji}$ and $M_{ii} = 0$ for $1 \leqslant i, j \leqslant k$. We thus have

$$\begin{aligned}E[M_\pi | \pi(1) = i] &= E[M(X_{\pi(1)}, X_{\pi(2)}) | \pi(1) = i] + \sum_{j=3}^{k} E[M(X_{\pi(j-1)}, X_{\pi(j)}) | \pi(1) = i] \\ &= \frac{\sum_{q \neq i} M_{iq}}{\binom{k-1}{1}} + (k-2)\frac{\sum_{p,q \neq i; p>q} M_{pq}}{\binom{k-1}{2}}.\end{aligned} \tag{6}$$

Hence, we can fix X_i at the first position of permutation π such that i maximizes expression (6). Notice that M_{ij} can be easily computed in polynomial time. Repeating the above process we can obtain a permutation π^* such that $M_{\pi*} \geqslant E[M_\pi]$. Hence, the next theorem follows.

Theorem 6. *There is a deterministic polynomial-time approximation algorithm for k-MCIP that achieves an asymptotic approximation ratio of* $0.5625k$.

6. Concluding remarks

In this paper, we proved a better approximation ratio for k-MCIP by viewing the problem as a flow decomposition problem in some flow network. We notice that our construction of the algorithm is simpler than that in [10] although the framework of our analysis is similar to that in [10]. It remains as an interesting question to approximate the minimum size of any set of good paths in a general flow network, not necessarily layered or acyclic.

Acknowledgment

We would like to thank David P. Woodruff for his comments on an earlier draft of this paper.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithm, and Applications, China Machine Press, 2005.
[2] X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong , S. Lonardi, T. Jiang, Computing the assignment of orthologous genes via genome rearrangement, in: Proc. Third Asia Pacific Bioinformatics Conf., 2005, pp. 363–378.
[3] X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong, S. Lonardi, T. Jiang, Assignment of orthologous genes via genome rearrangement, IEEE/ACM Trans. Comput. Biol. Bioinform. 2–4 (2005) 302–315.
[4] X. Chen, L. Liu, Z. Liu, T. Jiang, On the minimum common integer partition problem. CIAC2006, LNCS, Vol. 3998, Springer, Berlin, 2006, pp. 236–247.
[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, second ed. The MIT Press, Cambridge, MA, 2001, p. 1017.
[6] Z. Fu, X. Chen, V. Vacic, P. Nan, Y. Zhong, T. Jiang, A parsimony approach to genome-wide ortholog assignment, in: Proc. 10th Annu. Internat. Conf. on Research in Computational Molecular Biology, Venice, Italy, April 2006, pp. 578–594.
[7] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, Cambridge, 1995.
[8] C.H. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, J. Comput. System Sci. 43 (1991) 425–440.
[9] L. Valinsky, A. Schupham, G.D. Vedova, Z. Liu, A. Figueroa, K. Jampachaisri, B. Yin, E. Bent, R. Mancini-Jones, J. Press, T. Jiang, J. Borneman, Oligonucleotide fingerprinting of ribosomal RNA genes, in: Molecular Microbial Ecology Manual, second ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004, pp. 569–585.
[10] D.P. Woodruff, Better approximation for the minimum common integer partition problem, Manuscript, also available at ⟨http://web.mit.edu/dpwood/www/⟩, 2006.