

A Variant of Multi-task n-vehicle Exploration Problem: Maximizing Every Processor's Average Profit*

Yang-yang Xu[†], Jin-chuan Cui[‡]

Abstract

We discuss a variant of multi-task n-vehicle exploration problem. Instead of requiring an optimal permutation of vehicles in every group, the new problem asks all vehicles in a group to arrive at a same destination. It can also be viewed as to maximize every processor's average profit, given n tasks, and each task's consume-time and profit. Meanwhile, we propose a new kind of partition problem in fractional form, and analyze its computational complexity. Moreover, by regarding fractional partition as a special case, we prove that the maximizing average profit problem is NP-hard when the number of processors is fixed and it is strongly NP-hard in general. At last, a pseudo-polynomial time algorithm for the maximizing average profit problem and the fractional partition problem is presented, thanks to the idea of the pseudo-polynomial time algorithm for the classical partition problem.

Key words Multi-task n-vehicle exploration problem (MTNVEP), Maximizing average profit (MAP), Fractional partition (FP)

1 Introduction

Let $A = \{a_i \in \mathbf{Z}^+ : i = 1, \dots, n\}$, and $B = \{b_i \in \mathbf{Z}^+ : i = 1, \dots, n\}$. The n-vehicle exploration problem was introduced in [15] and analyzed in [8, 9, 13] to solve a scheduling

*Supported by 973 Program of China under Grant No. 2006CB701306, Key Laboratory of Management, Decision and Information Systems, CAS, and Beijing Research Center of Urban System Engineering.

[†]Email:xuyangyang@amss.ac.cn

[‡]Email:cjc@amss.ac.cn

problem

$$\max_{\pi} \frac{a_{\pi(1)}}{b_{\pi(1)}} + \frac{a_{\pi(2)}}{b_{\pi(1)} + b_{\pi(2)}} + \cdots + \frac{a_{\pi(n)}}{b_{\pi(1)} + \cdots + b_{\pi(n)}},$$

where $(\pi(1), \pi(2), \dots, \pi(n))$ is a permutation of $(1, 2, \dots, n)$. It can be described as^[15]: given n vehicles, the i th one can carry at most a_i liters of oil, and consumes b_i liters of oil per kilometer, $i = 1, \dots, n$. These vehicles start to travel towards one direction from the same position at the same time. On the path, they can not get oil from outside, but at any position any one can stop and transfer its left oil to other vehicles. How can we arrange these vehicles to make one of them travel farthest and ensure that all of them return to the original position?

The authors of [14] extended the n -vehicle exploration problem into a multi-task n -vehicle exploration problem (MTNVEP). They first divide n vehicles into m groups and then solve m new n -vehicle exploration problem to make every group travel far enough. That is to solve

$$\max \min_{1 \leq j \leq m} \max_{\pi_j} \frac{a_{\pi_j(1)}}{b_{\pi_j(1)}} + \frac{a_{\pi_j(2)}}{b_{\pi_j(1)} + b_{\pi_j(2)}} + \cdots + \frac{a_{\pi_j(n_j)}}{b_{\pi_j(1)} + \cdots + b_{\pi_j(n_j)}}, \quad (1)$$

where $(1_1, \dots, 1_{n_1}), \dots, (m_1, \dots, m_{n_m})$ is a partition of $(1, 2, \dots, n)$, and $(\pi_j(1), \dots, \pi_j(n_j))$ is a permutation of $(j_1, \dots, j_{n_j}), j = 1, \dots, m$. Note that the first ‘‘max’’ objective of (1) is about $(1, 2, \dots, n)$ ’s partition. They concluded that MTNVEP is NP-hard when m is fixed, and strongly NP-hard for general m . We notice that in their proof MTNVEP’s computational complexity is not related to vehicles’ permutation. Namely, even if the n -vehicle exploration problem can be solved in a polynomial time, MTNVEP is still NP-hard.

In this paper, we drop the permutation requirement of MTNVEP, but ask all vehicles in every group to arrive at a same destination. That is to solve

$$\max \min_{1 \leq j \leq m} \frac{a_{j_1} + \cdots + a_{j_{n_j}}}{b_{j_1} + \cdots + b_{j_{n_j}}}, \quad (2)$$

where $(1_1, \dots, 1_{n_1}), \dots, (m_1, \dots, m_{n_m})$ is a partition of $(1, 2, \dots, n)$. The ‘‘max’’ objective of (2) is still about $(1, \dots, n)$ ’s partition. This is a variant of multi-task n -vehicle exploration problem, which can be viewed as to maximize every processor’s average profit (MAP). It can be described as: given n tasks, and m identical processors, it will take b_i units of time to finish i th task and make a_i units of profit. How can we distribute the n tasks to m processors so that every processor can make its average profit large enough?

Meanwhile, we define a new kind of partition problem in fractional form (FP), if we change our objective to determine whether there exists a partition of $(1, 2, \dots, n)$ into

$(1_1, \dots, 1_{n_1}), \dots, (m_1, \dots, m_{n_m})$ such that

$$\frac{a_{j_1} + \dots + a_{j_{n_j}}}{b_{j_1} + \dots + b_{j_{n_j}}} = \frac{a_1 + a_2 + \dots + a_n}{b_1 + b_2 + \dots + b_n}, \text{ for any } j = 1, \dots, m,$$

instead of solving (2). FP is very similar to the classical partition problem proposed by Karp in his twenty-one NP-complete problems^[6], and 3-partition problem proposed by Garey and Johnson^[2,3]. The classical partition problem locates at the center of computationally intractable problems. Based on the partition problem, many other hard problems are proved to be NP-complete or NP-hard, such as bin packing^[4], multiprocessor scheduling problem^[5], and 0-1 integer programming^[11]. 3-Partition is often employed to devise a strong NP-completeness proof. We will prove FP's NP-completeness by reducing partition problem to FP when m is fixed, and prove FP's strong NP-completeness by reducing 3-partition to FP for general m . Meanwhile, we will prove that MAP is NP-hard^[3,12] for fixed $m \geq 2$, and strongly NP-hard for general $m \geq 2$ regarding FP as a special case of MAP. Moreover, thanks to the idea of designing pseudo-polynomial time algorithm for classical partition problem^[3,7], we design a pseudo-polynomial time algorithm for MAP and FP.

The outline of this paper is as follows: In Section 2, we give a general description of MAP and FP. Their complexity is analyzed in Section 3. We prove that FP is NP-complete and MAP is NP-hard for a fixed $m \geq 2$, and that FP is strongly NP-complete and MAP is strongly NP-hard for general $m \geq 2$. In Section 4, a pseudo-polynomial time algorithm for MAP and FP is presented, thanks to the idea of the pseudo-polynomial time algorithm for the partition problem. Section 5 concludes the paper.

2 The models

Definition 1 (Maximizing average profit). *Given $A = \{a_i \in \mathbf{Z}^+ : i = 1, \dots, n\}$, and $B = \{b_i \in \mathbf{Z}^+ : i = 1, \dots, n\}$, partition the index set $I = \{1, 2, \dots, n\}$ into m disjoint subsets I_1, I_2, \dots, I_m , denote*

$$r_j = \frac{\sum_{i \in I_j} a_i}{\sum_{i \in I_j} b_i}, \quad j = 1, \dots, m,$$

and let

$$f(I_1, I_2, \dots, I_m) = \min_{1 \leq j \leq m} r_j.$$

Determine the maximum value of f .

Definition 2 (Fractional partition). Given $A = \{a_i \in \mathbf{Z}^+ : i = 1, \dots, n\}$, $B = \{b_i \in \mathbf{Z}^+ : i = 1, \dots, n\}$, $I = \{1, 2, \dots, n\}$, and $S = \sum_{i \in I} a_i$, $T = \sum_{i \in I} b_i$, partition I into m disjoint subsets I_1, I_2, \dots, I_m , and denote

$$r_j = \frac{\sum_{i \in I_j} a_i}{\sum_{i \in I_j} b_i}, \quad j = 1, \dots, m.$$

Determine whether there exists a partition of I such that

$$r_j = \frac{S}{T}, \quad j = 1, \dots, m.$$

3 Complexity analysis

In the first subsection, we will prove that: for fixed $m \geq 2$, FP is NP-complete and MAP is NP-hard; in the second subsection, we will prove that: for general $m \geq 2$, FP is strongly NP-complete and MAP is strongly NP-hard.

3.1 NP-completeness

Theorem 1. *FP is NP-complete for fixed $m \geq 2$.*

Proof. At first, we prove this theorem for $m = 2$ in three steps.

Step 1. It is easy to see that FP belongs to NP, because given a partition of I , we can immediately calculate r_j and check whether $r_j = S/T$, for $j = 1, \dots, m$.

Step 2. We reduce one instance of partition problem to FP in a polynomial time.

The instance of partition problem is:

Q_1 : Given $C = \{c_i \in \mathbf{Z}^+, i = 1, \dots, n\}$, and $2K = \sum_i c_i$, determine whether there exists a subset $C_1 \subseteq C$, such that

$$\sum_{c \in C_1} c = \sum_{c \in C - C_1} c = K.$$

We construct FP's instance as follows:

Q_2 : Let

$$A = \left\{ Mc_1, \dots, Mc_n, \underbrace{M\delta, \dots, M\delta}_{N+n-1}, \underbrace{3M\delta/2, \dots, 3M\delta/2}_{N-2n+1}, MK, MK \right\}$$

$$B = \left\{ \underbrace{M, \dots, M}_{2N}, MN + M\epsilon, MN + M\epsilon \right\}$$

where

$$N = 4K + 1, \quad \epsilon = 1/\lceil 1/\epsilon' \rceil, \quad M = (5N - 4n + 1)/\epsilon, \quad \delta = \frac{2\epsilon}{5N - 4n + 1}$$

$$\epsilon' = \frac{-(4N^3 + 2KN - N^2 - 1) + \sqrt{(4N^3 + 2KN - N^2 - 1)^2 + 16N^3}}{8N^2};$$

Determine whether there exists a partition of the index set $I(= \{1, 2, \dots, 2N + 2\})$ into two disjoint subsets I_1, I_2 , such that

$$r_1 = r_2 = \frac{2K + \epsilon/2}{2N + \epsilon}.$$

It is easy to see that the reduction can be finished in a polynomial time. Though N, M , and K are not bounded by any polynomial of n , all numbers can be obtained in polynomial time. Moreover, we need construct $M\delta$ and $3M\delta/2$ only once, and then denote the number of times they appear in A by $N + n - 1$ and $N - 2n + 1$, respectively.

Note that all numbers in A and B have a common divisor M . This is to satisfy the integer requirement. Since the numerator and denominator of every fraction will have a common divisor M , it will not change the result to eliminate M in every number at the beginning. Thus we will consider the following sets

$$A = \left\{ c_1, \dots, c_n, \underbrace{\delta, \dots, \delta}_{N+n-1}, \underbrace{3\delta/2, \dots, 3\delta/2}_{N-2n+1}, K, K \right\}$$

$$B = \left\{ \underbrace{1, \dots, 1}_{2N}, N + \epsilon, N + \epsilon \right\}$$

Step 3. We prove that Q_1 is true if and only if Q_2 is true, where Q_i ($i = 1, 2$) is true if there exists a partition as described.

If Q_1 is true, i.e., there exists a subset $C_1 \subseteq C$, such that

$$\sum_{c \in C_1} c = \sum_{c \in C - C_1} c = K,$$

denote $n_1 = |C_1|$, $n_2 = n - n_1$, and set

$$I_1 = \{i : c_i \in C_1\} \cup \{n + 1, \dots, \frac{N-1}{2} + 3n_2, N + 2n, \dots, \frac{3N-1}{2} + 2n_1, 2N + 1\}$$

$$I_2 = I - I_1.$$

It is not difficult to verify that $r_1 = r_2 = (2K + \epsilon/2)/(2N + \epsilon)$. Namely, Q_2 is true.

Conversely, if Q_2 is true, i.e., there exists I 's partition I_1, I_2 , such that

$$r_1 = r_2 = \frac{2K + \epsilon/2}{2N + \epsilon}, \quad (3)$$

assume $\sum_{i \in I_j} c_i = p_j$, the numerator of fraction r_j is $p_j + y_j\delta + x_jK$, and the denominator of r_j is $q_j + x_j(N + \epsilon)$, where $x_j, y_j, q_j (j = 1, 2)$ are nonnegative integers. Then we get

$$r_j = \frac{p_j + y_j\delta + x_jK}{q_j + x_j(N + \epsilon)} = \frac{2K + \epsilon/2}{2N + \epsilon}, \quad j = 1, 2. \quad (4)$$

Note that $q_j \neq 0$; otherwise $p_j = y_j = 0$, so $r_j = K/(N + \epsilon)$. A contradiction to (3). Next we will prove

$$p_j = K, \quad j = 1, 2.$$

First, it must be $p_j/q_j \geq K/N$, $j = 1, 2$; otherwise, without loss of generality, we assume $p_1/q_1 < K/N$, then we have (because $q_1 \leq 2N$)

$$\frac{p_1}{q_1} \leq \frac{K}{N} - \frac{1}{2N^2}. \quad (5)$$

Transforming (4), we obtain

$$\frac{p_1}{q_1} = \frac{2K + \epsilon/2}{2N + \epsilon} + \frac{Nx_1\epsilon/2 + x_1\epsilon(K + \epsilon/2)}{q_1(2N + \epsilon)} - \frac{\epsilon y_1\delta + 2Ny_1\delta}{q_1(2N + \epsilon)}. \quad (6)$$

Using $N = 4K + 1 \geq 2n$ and $\delta = 2\epsilon/(5N - 4n + 1)$, it follows from (5) and (6) that

$$\frac{K}{N} - \frac{1}{2N^2} \geq \frac{p_1}{q_1} \geq \frac{2K + \epsilon/2}{2N + \epsilon} - \epsilon. \quad (7)$$

We can choose a sufficiently small $\epsilon > 0$ such that

$$\frac{K}{N} - \frac{1}{2N^2} < \frac{2K + \epsilon/2}{2N + \epsilon} - \epsilon.$$

That is

$$2N^2\epsilon^2 + (4N^3 + 2KN - N^2 - 1)\epsilon - 2N < 0.$$

If we choose ϵ' as half of the positive root of equation

$$2N^2\epsilon^2 + (4N^3 + 2KN - N^2 - 1)\epsilon - 2N = 0,$$

i.e.

$$\epsilon' = \frac{-(4N^3 + 2KN - N^2 - 1) + \sqrt{(4N^3 + 2KN - N^2 - 1)^2 + 16N^3}}{8N^2},$$

and let $\epsilon = 1/\lceil 1/\epsilon' \rceil$, then

$$\frac{K}{N} - \frac{1}{2N^2} < \frac{2K + \epsilon/2}{2N + \epsilon} - \epsilon.$$

A contradiction to (7). Thus $p_1/q_1 \geq K/N$, and $p_2/q_2 \geq K/N$.

Secondly, because $(p_1 + p_2)/(q_1 + q_2) = K/N$, we get

$$\frac{p_j}{q_j} = \frac{K}{N}, \quad j = 1, 2. \quad (8)$$

Write (8) as $2p_jN = 2q_jK$. Note that N and $2K$ are relatively prime^[1], so q_j is divisible by N . That is, $q_j = N$ or $2N$, but $q_1 + q_2 = 2N$, thus $q_j = N$, $j = 1, 2$. Together with (8), we have

$$p_j = K, \quad j = 1, 2.$$

If we set

$$C_1 = \{c_i \in C \mid i \in I_1\},$$

then

$$\sum_{c \in C_1} c = \sum_{c \in C - C_1} c = p_1 = K.$$

Namely, Q_1 is true.

Until now, we have proved that FP is NP-complete when $m = 2$; for $m > 2$, we can construct FP's instance as follows.

Q'_2 : Let

$$A = \left\{ Mc_1, \dots, Mc_n, \underbrace{M\delta, \dots, M\delta}_{N+n-1}, \underbrace{3M\delta/2}_{N-2n+1}, MK, MK, \underbrace{2MK + M\epsilon/2}_{m-2} \right\}$$

$$B = \left\{ \underbrace{M, \dots, M}_{2N}, MN + M\epsilon, MN + M\epsilon, \underbrace{2MN + M\epsilon}_{m-2} \right\}$$

where

$$N = 4K + 1, \quad \epsilon = 1/\lceil 1/\epsilon' \rceil, \quad M = (5N - 4n + 1)/\epsilon, \quad \delta = \frac{2\epsilon}{5N - 4n + 1}$$

$$\epsilon' = \frac{-(4N^3 + 2KN - N^2 - 1) + \sqrt{(4N^3 + 2KN - N^2 - 1)^2 + 16N^3}}{8N^2};$$

Determine whether there exists a partition of the index set $I(= \{1, 2, \dots, 2N + m\})$ into m disjoint subsets I_1, I_2, \dots, I_m , such that

$$r_j = \frac{2K + \epsilon/2}{2N + \epsilon}, \quad j = 1, \dots, m.$$

This construction can also be finished in polynomial time. The proof of equivalence between Q_1 and Q'_2 is essentially the same as that when $m = 2$.

Therefore, FP is NP-complete for fixed $m \geq 2$. \square

Theorem 2. *MAP is NP-hard for fixed $m \geq 2$.*

Proof. Given $A = \{a_i \in \mathbf{Z}^+ : i = 1, \dots, n\}$, $B = \{b_i \in \mathbf{Z}^+ : i = 1, \dots, n\}$, $I = \{1, 2, \dots, n\}$, and $S = \sum_{i \in I} a_i$, $T = \sum_{i \in I} b_i$, partition I into m disjoint subsets I_1, I_2, \dots, I_m , and denote

$$r_j = \frac{\sum_{i \in I_j} a_i}{\sum_{i \in I_j} b_i}, \quad j = 1, \dots, m.$$

Then the proof of this theorem is straightforward from “ $r_j \geq S/T$ for any j if and only if $r_j = S/T$ for any j .” \square

3.2 Strong NP-completeness

Theorem 3. *FP is strongly NP-complete for general $m \geq 2$.*

Proof. In last subsection, we have shown that FP belongs to NP. Next, we only need reduce 3-Partition to FP for general $m \geq 2$ in pseudo-polynomial time.

An instance of 3-Partition is:

Q_3 : Given $D = \{d_i \in \mathbf{Z}^+ : i = 1, \dots, 3m\}$, $mK = \sum_i d_i$, and $K/4 < d_i < K/2$ for every i , determine whether there exists a partition of D into m disjoint subsets D_1, \dots, D_m , such that

$$\sum_{d \in D_j} d = K, \quad j = 1, \dots, m.$$

We construct FP's instance as follows:

Q_4 : Let

$$A = \left\{ Md_1, \dots, Md_{3m}, \underbrace{M\delta}_{mN-3m}, \underbrace{MK}_m \right\}$$

$$B = \left\{ \underbrace{M, \dots, M}_{mN}, \underbrace{MN + M\epsilon}_m \right\}$$

where

$$N = 2mK + 1, \quad \epsilon = 1/\lceil 1/\epsilon' \rceil, \quad M = (mN - 3m)/\epsilon, \quad \delta = \frac{\epsilon}{mN - 3m}$$

$$\epsilon' = \frac{-(2mN^3 + mKN - 1) + \sqrt{(2mN^3 + mKN - 1)^2 + 8mN^3}}{4mN^2};$$

Determine whether there exists a partition of the index set $I(= \{1, 2, \dots, m(N + 1)\})$ into m disjoint subsets I_1, I_2, \dots, I_m , such that

$$r_j = \frac{2K + \epsilon/m}{2N + \epsilon}, \quad j = 1, \dots, m.$$

We bring a common divisor M in each element of A and B , for the same reason as we proved theorem 1. It will not change the result to eliminate M at the beginning. Thus we will consider the following sets

$$A = \left\{ d_1, \dots, d_{3m}, \underbrace{\delta}_{mN-3m}, \underbrace{K}_m \right\}, \quad B = \left\{ \underbrace{1, \dots, 1}_{mN}, \underbrace{N + \epsilon}_m \right\}.$$

Now we focus on proving that Q_3 is true if and only if Q_4 is true.

If Q_3 is true, i.e., there exists a partition of D into m disjoint subsets D_1, \dots, D_m , such that

$$\sum_{d \in D_j} d = K, \quad j = 1, \dots, m,$$

there are exactly three elements in every subset, since $K/4 < d_i < K/2$, $i = 1, \dots, 3m$. Letting

$$I_j = \{i : d_i \in D_j\} \cup \{3m + (j-1)(N-3) + 1, \dots, 3m + j(N-3), mN + j\},$$

we have

$$r_j = \frac{2K + (N-3)\delta}{2N + \epsilon} = \frac{2K + \epsilon/m}{2N + \epsilon}, \quad j = 1, \dots, m.$$

Namely, Q_4 is true.

Conversely, if Q_4 is true, i.e., there exists a partition of the index set I into m disjoint subsets I_1, \dots, I_m , such that

$$r_j = \frac{2K + \epsilon/m}{2N + \epsilon}, \quad j = 1, \dots, m, \quad (9)$$

assume $\sum_{i \in I_j} d_i = p_j$, the numerator of r_j is $p_j + x_j K + y_j \delta$, and the denominator of r_j is $q_j + x_j(N + \epsilon)$, where $x_j, y_j, q_j (j = 1, 2)$ are nonnegative integers. Then we have

$$r_j = \frac{p_j + x_j K + y_j \delta}{q_j + x_j(N + \epsilon)} = \frac{2K + \epsilon/m}{2N + \epsilon}, \quad j = 1, \dots, m.$$

Note that $q_j \geq 1$ for every $j = 1, \dots, m$; otherwise $p_j = y_j = 0$, then $r_j = K/(N + \epsilon)$. A contradiction to (9). Next, we will prove

$$p_j = K, \quad j = 1, \dots, m$$

Firstly, it must be $p_j/q_j \geq K/N$, $j = 1, \dots, m$. Otherwise, without loss of generality, we assume $p_1/q_1 < K/N$, then we have (because $q_1 \leq mN$)

$$\frac{p_1}{q_1} \leq \frac{K}{N} - \frac{1}{mN^2}.$$

Because

$$r_1 = \frac{p_1 + x_1 K + y_1 \delta}{q_1 + x_1(N + \epsilon)} \leq \frac{p_1 + x_1 K + \epsilon}{q_1 + x_1(N + \epsilon)}$$

and

$$r_1 = \frac{2K + \epsilon/m}{2N + \epsilon} > \frac{2K}{2N + \epsilon},$$

we have

$$\frac{p_1 + x_1K + \epsilon}{q_1 + x_1(N + \epsilon)} > \frac{2K}{2N + \epsilon}. \quad (10)$$

Transforming (10), we get

$$\begin{aligned} \frac{p_1}{q_1} &> \frac{2K}{2N + \epsilon} + \frac{\epsilon x_1 K}{q_1(2N + \epsilon)} - \frac{\epsilon}{q_1} \\ &\geq \frac{2K}{2N + \epsilon} + \frac{\epsilon x_1 K}{q_1(2N + \epsilon)} - \epsilon \\ &\geq \frac{2K}{2N + \epsilon} - \epsilon. \end{aligned} \quad (11)$$

We can choose a sufficiently small $\epsilon > 0$, such that

$$\frac{K}{N} - \frac{1}{mN^2} < \frac{2K}{2N + \epsilon} - \epsilon;$$

that is

$$mN^2\epsilon^2 + (2mN^3 + mKN - 1)\epsilon - 2N < 0.$$

If we choose ϵ' as half of the positive root of equation

$$mN^2\epsilon^2 + (2mN^3 + mKN - 1)\epsilon - 2N = 0,$$

i.e.

$$\epsilon' = \frac{-(2mN^3 + mKN - 1) + \sqrt{(2mN^3 + mKN - 1)^2 + 8mN^3}}{4mN^2},$$

and let $\epsilon = 1/\lceil 1/\epsilon' \rceil$, then

$$\frac{K}{N} - \frac{1}{mN^2} < \frac{2K}{2N + \epsilon} - \epsilon.$$

But

$$\frac{p_1}{q_1} \leq \frac{K}{N} - \frac{1}{mN^2},$$

a contradiction to (11). Thus, for the chosen ϵ , it must be

$$\frac{p_j}{q_j} \geq \frac{K}{N}, \quad j = 1, \dots, m.$$

Secondly, because

$$\frac{p_j}{q_j} \geq \frac{K}{N}, \quad j = 1, \dots, m,$$

and

$$\frac{\sum p_j}{\sum q_j} = \frac{mK}{mN} = \frac{K}{N},$$

we get

$$\frac{p_j}{q_j} = \frac{K}{N}, \quad j = 1, \dots, m. \quad (12)$$

Write (12) as $mp_jN = mq_jK$, $j = 1, \dots, m$. Note that N and mK are relatively prime, thus q_j is divisible by N . Because $\sum_{j=1}^m q_j = mN$, we get $q_j = N$, $j = 1, \dots, m$. Together with (12), we have

$$p_j = K, \quad j = 1, \dots, m.$$

If we set

$$D_j = \{d_i \in D : i \in I_j\}, \quad j = 1, \dots, m,$$

then

$$\sum_{d \in D_j} d = p_j = K, \quad j = 1, \dots, m.$$

Namely, Q_3 is true.

Therefore, FP is strongly NP-complete for general $m \geq 2$. □

Theorem 4. *MAP is strongly NP-hard for general $m \geq 2$.*

Proof. The proof of this theorem is essentially the same as that of theorem 2. □

4 Pseudo-polynomial algorithm

In last section, we analyzed the computational complexity of FP and MAP: FP is NP-complete and MAP is NP-hard for fixed $m \geq 2$; FP is strongly NP-complete and MAP is strongly NP-hard for general $m \geq 2$. According to [3], if problem Π is strongly NP-complete, then it is impossible to solve Π in a pseudo-polynomial time unless $P=NP$. As a result, we do not hope to find FP and MAP's pseudo-polynomial time algorithm for general $m \geq 2$. However, we can prove that FP and MAP can be solved in a pseudo-polynomial time when m is fixed. The pseudo-polynomial time algorithm we will present for FP and MAP is similar to that for the classical partition problem.

$$\left[\begin{array}{c} t(1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{0}{0}), t(1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{0}{1}), \dots, t(1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{0}{T}) \\ t(1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{1}{0}), t(1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{1}{1}), \dots, t(1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{1}{T}) \\ \dots \\ t(1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{S}{0}), t(1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{S}{1}), \dots, t(1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{S}{T}) \end{array} \right] \dots$$

$$\left[\begin{array}{c} t(n-1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{0}{0}), t(n-1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{0}{1}), \dots, t(n-1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{0}{T}) \\ t(n-1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{1}{0}), t(n-1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{1}{1}), \dots, t(n-1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{1}{T}) \\ \dots \\ t(n-1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{S}{0}), t(n-1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{S}{1}), \dots, t(n-1, \frac{S}{T}, \dots, \frac{S}{T}, \frac{S}{T}) \end{array} \right]$$

While listing these matrices, if some

$$t(i, \frac{p_1}{q_1}, \dots, \frac{p_{m-1}}{q_{m-1}}) = 1,$$

and

$$\frac{p_j}{q_j} = \frac{S}{T}, \quad j = 1, \dots, m-1$$

$$\frac{S - \sum_{j=1}^{m-1} p_j}{T - \sum_{j=1}^{m-1} q_j} = \frac{S}{T},$$

the algorithm stops and return “FP is true”; else return “FP is false”.

For example, when $m = 2$, we can list

$$\left[\begin{array}{c} t(1, \frac{0}{0}), t(1, \frac{0}{1}), \dots, t(1, \frac{0}{T}) \\ t(1, \frac{1}{0}), t(1, \frac{1}{1}), \dots, t(1, \frac{1}{T}) \\ \dots \\ t(1, \frac{S}{0}), t(1, \frac{S}{1}), \dots, t(1, \frac{S}{T}) \end{array} \right] \dots \left[\begin{array}{c} t(i, \frac{0}{0}), t(i, \frac{0}{1}), \dots, t(i, \frac{0}{T}) \\ t(i, \frac{1}{0}), t(i, \frac{1}{1}), \dots, t(i, \frac{1}{T}) \\ \dots \\ t(i, \frac{S}{0}), t(i, \frac{S}{1}), \dots, t(i, \frac{S}{T}) \end{array} \right]$$

$$\dots \left[\begin{array}{c} t(n-1, \frac{0}{0}), t(n-1, \frac{0}{1}), \dots, t(n-1, \frac{0}{T}) \\ t(n-1, \frac{1}{0}), t(n-1, \frac{1}{1}), \dots, t(n-1, \frac{1}{T}) \\ \dots \\ t(n-1, \frac{S}{0}), t(n-1, \frac{S}{1}), \dots, t(n-1, \frac{S}{T}) \end{array} \right]$$

In all matrices, there are $(n - 1)(ST)^{m-1}$ entries in total, and to get the value of $t(i, p_1/q_1, \dots, p_{m-1}/q_{m-1})$ needs at most m steps, so the computational complexity is $O(mn(ST)^{m-1})$. Therefore, FP can be solved in a pseudo-polynomial time. \square

In the same way, we can solve MAP in a pseudo-polynomial time.

Theorem 6. *MAP can be solved in a pseudo-polynomial time for fixed $m \geq 2$.*

Proof. Define

$$g(i, \frac{p_1}{q_1}, \dots, \frac{p_{m-1}}{q_{m-1}}) = \begin{cases} 0 & \text{if } t(\cdot) = 0 \\ \min(\frac{p_1}{q_1}, \dots, \frac{p_{m-1}}{q_{m-1}}, \frac{S - \sum_{j=1}^{m-1} p_j}{T - \sum_{j=1}^{m-1} q_j}) & \text{if } t(\cdot) = 1 \end{cases}$$

While listing matrices in the proof of theorem 5, we record the value of function g together with the boolean function t . At the end, we search for all entries in these matrices to obtain the maximum value of g , which is the optimal value of MAP. To get the value of $g(i, p_1/q_1, \dots, p_{m-1}/q_{m-1})$ needs at most m steps, and to search for all entries needs $(n - 1)(ST)^{m-1}$ steps, so the computational complexity is $O(mn(ST)^{m-1})$. Therefore, MAP can be solved in pseudo-polynomial time. \square

5 Conclusion and future work

The multi-task n-vehicle exploration problem is very important in the context of how to use limited resources to complete a number of tasks. Due to the fact that MTNVEP is still NP-hard even if the n-vehicle exploration problem can be solved in polynomial time, we try to drop the permutation requirement but ask all vehicles in one group to arrive at a same destination. The new problem is a variant of multi-task n-vehicle exploration problem. It can also be regarded as how to maximize every processor's average profit. Through analyzing its special case: fractional partition, we prove that MAP is NP-hard when the processor number is fixed, and it is strongly NP-hard in general.

Fractional partition is a new kind of partition problem with a structure similar to the classical partition problem and 3-partition. To the best of our knowledge, this is the first paper about fractional partition. Stancu-Minasian gives a large quantity of fractional programming problems^[10], but all of them are optimization problems and their constraints are not fractional. In fact, we find MAP can be modeled as a fractional programming problem. Not only the objective function will be fractional, but also the constraints are fractional. Some approximation algorithms will be presented in our subsequent papers through modeling MAP as a fractional programming problem.

References

- [1] Dummit, D. S., Foote, R. M. Abstract Algebra, Prentice-Hall, New Jersey, 1999
- [2] Garey, M. R., Johnson, D. S. Complexity results for multiprocessor scheduling under resource constraints. *Siam Journal on Computing*, 4(4): 397-411 (1975)
- [3] Garey, M. R., Johnson, D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, 1979
- [4] Garey, M. R., Johnson, D. S. A 71/60 theorem for bin packing. *J. Complexity* 1(1): 65-106 (1985)
- [5] Graham, R. L. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal* 45(9): 1563-1581 (1966)
- [6] Karp. Reducibility among combinatorial problems. in "R. E. Miller, J. W. Thatcher (Eds), Complexity of Computer Computations: proceedings of a symposium on the Complexity of Computer Computation", pp.85-103, Plenum, New York, 1972
- [7] Kleinberg, J., Tardos, E. Algorithm Design, Addison Wesley, United States ED edition, 2005
- [8] Li, X. Y., Cui, J. C. Efficient algorithm for a kind of exploration problem with n vehicles. *J. System Engineering*, 8: 444-448 (2008)
- [9] Liang, L. L. A problem of permutation optimization. *J. Guangxi University for Nationalities*, 12(4): 72-76 (2006)
- [10] Stancu-Minasian, I. M. Fractional Programming: theory, methods and applications, Kluwer Academic Publishers, Boston, 1997
- [11] Taha, H. A. Integer Programming: theory, applications, and computations, Academic Press, New York, 1975
- [12] Wegener, I. Complexity Theory, Science Press, Beijing, 2006
- [13] Xia, X., Cui, J. C. A Method of Estimating Computational Complexity Based on Input Conditions for N-vehicle Problem. *Acta Mathematicae Applicatae Sinica* (English series), 26(1): 1-12 (2010)
- [14] Xu, Y. Y., Cui, J. C. Multitask N-Vehicle Exploration Problem: complexity and algorithm. (working paper)

- [15] Yang, R. A traffic problem with oil and its extension. *Mathematical communication*, 9: 44-45 (1999)