Journal of Combinatorial Optimization, 6, 67–80, 2002

3-Partitioning Problems for Maximizing the Minimum Load*,†

SHI PING CHEN
Department of Mathematics, Zhejiang University, Hangzhou 310027, People's Republic of China

YONG HE heyong@math.zju.edu.cn
Department of Mathematics, Zhejiang University, Hangzhou 310027, People's Republic of China

GUOHUI LIN ghlin@cs.ualberta.ca
Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2E8; Department of Computer Science, University of Waterloo; Department of Computing and Software, McMaster University

Received October 12, 1999; Revised February 24, 2000; Accepted February 25, 2000

Abstract. The optimization versions of the 3-PARTITIONING and the KERNEL 3-PARTITIONING problems are considered in this paper. For the objective to maximize the minimum load of the m subsets, it is shown that the MODIFIED LPT algorithm has performance ratios $(3m-1)/(4m-2)$ and $(2m-1)/(3m-2)$, respectively, in the worst case.

Keywords: partitioning, scheduling, analysis of algorithm, worst case performance ratio, kernel

1. Introduction

Set partitioning problems generally ask for a partition of a given set of positive real numbers into a given number of subsets such that the sums of elements in the subsets are as nearly equal as possible. 3-PARTITIONING is one of the basic NP-complete problems (Garey and Johnson, 1978), in which $3m$ elements have to be partitioned into m subsets each of which contains three elements. In this paper we consider the following generalized version:

> Given a set A of n positive numbers, i.e., $A = \{p_1, p_2, \ldots, p_n\}$, $n \leq 3m$, we look for a partition of A into m subsets such that each subset can contain up to three elements and the sums of elements in the subsets (called *loads*) are "nearly" equal.

To achieve the near-equality, one may in one way minimize the maximum load of the m subsets (i.e., *makespan*), or in another way maximize the minimum load of the m subsets.

*This research is supported by National Natural Science Foundation of China (grant 19701028) and 973 National Fundamental Research Project of China.
†Supported in part by NSERC Research Grant OGP0046613 and a CITO grant.

For the first objective, Kellerer and Woeginger (1993) presented a MODIFIED LPT algorithm (denoted by MLPT in the following). At the beginning, all the m subsets are *open* to receive elements. The MLPT assigns iteratively the largest unassigned element to an open subset with the least current load. A subset which contains three elements is *closed* to which no more element can be assigned by the algorithm. The algorithm terminates till every element has been assigned to some subset. It is shown that the MLPT has a tight performance ratio $4/3 - 1/3m$. Later, Kellerer and Kotov devised a better approximation algorithm with a worst-case performance ratio $7/6$. Recently, Babel et al. (1998) investigated the general k-PARTITIONING problem where $k \geq 3$. They devised an approximation algorithm with a worst-case performance ratio 4/3. In this paper, we investigate 3-PARTITIONING under the second objective. We will show that the MLPT has a worst-case performance ratio $(3m - 1)/(4m - 2)$.

Chen et al. (1996) proposed to study a variant of 3-PARTITIONING, called KERNEL 3-PARTITIONING. It can be described as follows:

> Let $A = \{g_1, g_2, \ldots, g_m, p_1, p_2, \ldots, p_n\}$ be a set of $m + n (n \leq 2m)$ elements, where each g_i is a *kernel* and it is a nonnegative number and each p_i is an ordinary element and it is a positive number. We look for a partition of A into m subsets such that (1) each subset contains exactly one Kernel, (2) each subset contains up to three elements, and (3) the loads of subsets are "nearly" equal.

It can be shown that KERNEL 3-PARTITIONING is NP-hard (Chen et al., 1996) as well. Following Kellerer and Woeginger (1993), Chen et al. (1996) considered the objective of minimizing the makespan. It was shown that the MLPT has a tight worst-case performance ratio $3/2 - 1/2m$. Here the MLPT first assigns the m kernels, one into a subset, then it assigns the ordinary elements as the above MLPT algorithm does. In this paper, we also investigate KERNEL 3-PARTITIONING to maximize the minimum load of the m subsets. We will show that the MLPT has a tight performance ratio $(2m - 1)/(3m - 2)$.

Strongly related to 3-PARTITIONING and KERNEL 3-PARTITIONING is the following fundamental problem in Scheduling Theory: Schedule n independent tasks non-preemptively on a multiprocessor system, where the tasks are all available at time zero and machine M_i $(i = 1, \ldots, m)$ is available at time g_i. The goal is to look for a schedule to minimize the makespan (Chang and Hwang, 1999; Lee, 1991; Lee et al., 2000), or maximize the minimum machine completion time (Lin et al., 1998). We denote these two problems as $P, g_i \| C_{max}$ and $P, g_i \| C_{min}$, respectively. If all g_i are zero, they become the classical scheduling problems initially proposed in Deuermeyer et al. (1982) and Graham (1969), which are denoted by $P \| C_{max}$ and $P \| C_{min}$, respectively, in the literature. $P \| C_{max}$ and $P \| C_{min}$ closely relate to 3-PARTITIONING; while $P, g_i \| C_{max}$ and $P, g_i \| C_{min}$ relate to KERNEL 3-PARTITIONING. To show their intimate relationship, Kellerer and Kotov gives an application of the approximation algorithms for 3-PARTITIONING to the corresponding scheduling problem $P \| C_{max}$. Babel et al. (1998) also showed the relationship between the scheduling problems and the k-PARTITIONING problem.

For most combinatorial problems, such as set partitioning and scheduling problems, it is an important job to get the worst-case performance guarantee of a greedy-like approximation algorithm, such as the LPT-like algorithms in Scheduling Theory. One of the reasons is the simplicity and the effectiveness of such algorithms. In fact, "LPT algorithm has been the touchstone for the design of efficient off-line algorithms" (Chen, 1994). Another reason may be theoretical since in many cases to get the worst-case performance ratio is not an easy task. A typical example is the LPT algorithm for the above multiprocessor scheduling problem. For $P\|C_{min}$, due to the much profound difficulty of the maximin criterion, the worst-case performance ratio of the LPT had not been worked out until 1992. Deuermeyer et al. (1982) initiated this work in 1982. Ten years later, Csirik et al. (1992) proved that the exact ratio is $(3m-1)/(4m-2)$. In 1998, Lin et al. showed that the worst-case performance ratio of the LPT for $P, g_i\|C_{min}$ is $(2m-1)/(3m-2)$. In this paper, we continue this line of work. We will prove the worst-case performance ratios of the MLPT by employing the methods such as *minimal counterexample, domination* and *weighting function*, etc. The proofs are very technical and thus on the other hand show again the power of these "traditional" methods. In addition to those traditional methods, we introduce a new technique—*enlarging* process—in the proof of nonexistence of a minimal counterexample. This process enlarges some elements in the given set to a specified value in order to get a contradiction. It is very powerful and may be of independent interest.

In the following two sections, we prove the tight ratios of the MLPT algorithm applying to 3-PARTITIONING and KERNEL 3-PARTITIONING, respectively, under the objective of maximizing the minimum load. Although the details of routines dealing with them are very different, the ideas are quite the same. Therefore, we are going to give the full details in Section 2, while Section 3 contains only the outline and some important "branching points". Before we start the main part, we list some useful notations: Let σ be the partition of A yielded by the MLPT, $\sigma = \{A_1, \dots, A_m\}$, and σ^* be an optimal partition of A, $\sigma^* = \{A_1^*, \dots, A_m^*\}$. As the MLPT partitioning process proceeds, denote C_i the sum of the elements already assigned to A_i at a certain time, and call it the load of A_i at that time. Denote C_i^H and C_i^* the final load of A_i and A_i^*, respectively. Denote $w = \min_{1\le i\le m}\{C_i^H\}$, and $w^* = \min_{1\le i\le m}\{C_i^*\}$.

2. Performance of the MLPT applying to 3-PARTITIONING

In this section, we consider the 3-PARTITIONING problem under the objective of maximizing the minimum load. More specifically, we investigate the performance of the MLPT algorithm. The whole section is devoted to proving

Theorem 2.1. *The tight performance ratio of the MLPT is* $(3m-1)/(4m-2)$.

Recall that the worst-case performance ratio of the LPT applying to $P\|C_{min}$ is also $(3m-1)/(4m-2)$ (Csirik et al., 1992). Theorem 2.1 states that the additional restriction of at most three elements per subset has no influence on the worst-case behavior of the MLPT.

Proof: The proof will be done by contradiction and hence we introduce first a *minimal counterexample*. An *m-counterexample* is an instance of 3-PARTITIONING in which set A contains $n \le 3m$ positive numbers to be partitioned into m subsets, and for which

$w/w^* < (3m-1)/(4m-2)$. (Note that there should be $m \geq 2$.) A *minimal counterexample* is an m-counterexample in the sense that the parameter m is the minimal, that is, no m'-counterexample exists with $m' < m$. Clearly, if the theorem doesn't hold, then there exists a counterexample. The existence of a counterexample implies the existence of a minimal counterexample. Thus, suppose the minimal counterexample is an m-counterexample and we can show that either it isn't a counterexample or there exists an $(m-1)$-counterexample, we will get a contradiction. Indeed we will do this in the following.

Let I denote this minimal counterexample. W.l.o.g., we assume that the elements of A are sorted as $p_1 \geq p_2 \geq \cdots \geq p_n$. If during the MLPT partitioning process, the case—that a closed subset has the least load at some time—doesn't happen, that is, the MLPT behaves just the same as the LPT for this particular instance, then we can draw the conclusion that for $I : w/w^* \geq (3m-1)/(4m-2)$ (Csirik et al., 1992). That means I is not a counterexample, and hence reaches the contradiction. So we can assume that in the MLPT partitioning process there is some time at which a closed subset has the least load—the MLPT cannot assign the element under consideration to it because it contains already three elements. Obviously, the load of this closed subset is exactly w.

It follows that if in I the number of elements $n < 3m$, then we may add $3m - n$ more elements each having a value equal to p_n into A. Adding these elements doesn't change the assignment of $\{p_1, p_2, \ldots, p_n\}$ by the MLPT and thus doesn't change the value w. On the other hand, the new value of w^* is equal to or larger than the old value of w^* since there are more elements to be partitioned. This tells us that w/w^* does not increase and hence I remains as a (minimal) counterexample. Thus, we may assume without loss of generality that in I the number of elements $n = 3m$. It then follows that in any partition of A every subset contains exactly three elements.

In the MLPT partition σ, we suppose the elements in A_i are p_{i_1}, p_{i_2} and p_{i_3} with the indices $i_1 < i_2 < i_3$, which are called the *first, second*, and *third* element of A_i, respectively. In the optimal partition σ^*, we suppose $A_i^* = \{p_{i_1^*}, p_{i_2^*}, p_{i_3^*}\}$ with the indices $i_1^* < i_2^* < i_3^*$ and call them the first, second, and third element of A_i^*, respectively, as well. Moreover, we normalize the elements of A in such a way that $w^* = 4 - 2/m$. It then follows that $w < 3 - 1/m$. Furthermore, since the sum of all elements is at least $4m - 2$, we derive that the makespan of σ is greater than $4 - 1/m$. □

Definition 2.1 (Csirik et al., 1992). A subset $A_i = \{p_{i_1}, p_{i_2}, p_{i_3}\}$ in σ is dominated by a subset $A_j^* = \{p_{j_1^*}, p_{j_2^*}, p_{j_3^*}\}$ in σ^*, if $p_{i_r} \leq p_{j_r^*}$, for $r = 1, 2, 3$.

We remark that if A_j^* dominates A_i, it is not necessary that the indices $j_r^* \leq i_r$ for $r = 1, 2, 3$. During the MLPT partitioning process, suppose A_k is the first subset which is assigned three elements and its load is $C_k^H = w$. For simplicity, denote $s = k_3$, the index of the third element in subset A_k. We enlarge all elements of A which are smaller than p_s to p_s. Notice that for the reason the same as above this enlarging process does not change the value of w, nor it decreases the value of w^*, and thus I remains as a (minimal) counterexample.

Lemma 2.1 (*Domination Lemma* (Csirik et al., 1992)). *For any $i \neq k$, there is no subset A_j^* in σ^* that would dominate subset A_i in σ.*

Proof: Suppose to the contrary that A_j^* dominates A_i, then we can get an $(m-1)$-counterexample I' by asking for a partition of set $A' = A \backslash A_i$ into $m-1$ subsets. The reasons that I' is a counterexample are: (1) The MLPT partition of A' is in fact identical to the MLPT partition of A excluding subset A_i; and (2) Deleting the subset A_j^* from the optimal partition of A and replacing element p_{i_r} by element $p_{j_r}^*$, for $r = 1, 2, 3$, will form a partition of A'. In this partition, the minimum load is at least as large as w^*. Therefore, the optimal partition of A' has also a minimum load at least as large as w^*. But this contradicts the minimality of the parameter m. Thus there shouldn't be any subset A_j^* dominating A_i. □

Lemma 2.2. *During the MLPT partitioning process, at the time p_s is assigned to subset A_k, each of the other subsets contains at least two elements.*

Proof: Suppose to the contrary, there is some subset A_i which contains only one (note that there should be at least one) element p_{i_1}. It is clear that $i \neq k$, and $p_{i_2} = p_{i_3} = p_s$. Assume in the optimal partition σ^*, p_{i_1} is assigned to A_j^* then A_j^* dominates A_i, a contradiction to Lemma 2.1. □

Corollary 2.1. $p_1 \leq w - p_s$.

Proof: Noticing that if $p_1 > w - p_s$, then at the time p_s is assigned to A_k, there is only one element, which is p_1, in the subset that contains p_1. This is a contradiction to Lemma 2.2. □

Let f denote the largest index among those m second elements of subsets in the MLPT partition σ, then $f < s$ by Lemma 2.2. Suppose p_f is assigned to subset A_l by the MLPT, i.e., $f = l_2$, then $p_{l_1} = p_1$ (note: not necessarily the index $l_1 = 1$). Similarly, we can enlarge all elements p_i's with $s > i > f$ to p_f.

Lemma 2.3. $p_{i_2} \leq (w - p_s)/2$, *for any* $i = 1, 2, \ldots, m$.

Proof: We will prove that $p_{m+1} \leq (w - p_s)/2$ in the following. The lemma then follows directly. If $p_{m+1} > (w - p_s)/2$, and suppose it is assigned to subset A_i in σ for some i, then $i \neq k$, $p_{i_1} = p_m$ (note: again not necessarily index $i_1 = m$) and the third element assigned to A_i is equal to p_s. Note that there is a subset in σ^*, say A_j^*, which contains at least two elements in $\{p_1, p_2, \ldots, p_{m+1}\}$. It follows that A_j^* dominates A_i, a contradiction. □

Since the makespan of the MLPT partition σ is greater than $4 - 1/m$, from Lemma 2.2 we know that there is some subset A_i whose load exceeds $4 - 1/m - p_s$ at the time p_s is assigned to A_k. Suppose that, during the MLPT partitioning process, A_u is the first subset with its load exceeding $4 - 1/m - p_s$ and it is the element p_x that assigning it to A_u makes the load exceed $4 - 1/m - p_s$. We note that element p_x might be the second or the third element in subset A_u. Now we trace back the partial MLPT partition obtained at the time right after p_x is assigned to A_u. Let C_u denote the load of A_u at that time, and $Q_0 = C_u - p_x$, then

$$Q_0 + p_x = C_u > 4 - \frac{1}{m} - p_s. \tag{2.1}$$

We will get the contradiction by distinguishing two cases according to the position of p_x in A_u.

2.1. *Case 1: p_x is the second element in A_u*

In this case, $Q_0 = p_{u_1}$. By Corollary 2.1 and Lemma 2.3,

$$1 < \left(4 - \frac{1}{m} - p_s\right) - (w - p_s) < p_x \leq \frac{w - p_s}{2} < \frac{3}{2} - \frac{1}{2m} - \frac{p_s}{2}. \tag{2.2}$$

Combining the latter part of (2.2) with (2.1), we derive that

$$Q_0 > \frac{5}{2} - \frac{1}{2m} - \frac{p_s}{2}. \tag{2.3}$$

Moreover, since $Q_0 = p_{u_1} \leq w - p_s$, we have

$$p_s < 1 - \frac{1}{m}. \tag{2.4}$$

The following lemma, which is an improvement of Corollary 2.1, holds in this case.

Lemma 2.4. $p_1 \leq w - p_s - p_f$.

Proof: Suppose to the contrary, then we have the first element of subset A_l:$p_{l_1} = p_1 > w - p_s - p_f$. Therefore, $l \neq k$ and $A_l = \{p_{l_1}, p_f, p_{l_3}\}$ where the third element p_{l_3} has a value equal to p_s. However, by Corollary 2.1 and (2.4), $p_{l_1} + 2p_s \leq (w - p_s) + 2p_s = w + p_s < w^*$. That means the subset A_j^* containing element p_1 contains another element which is larger than or equal to p_f. It then follows that A_l is dominated by subset A_j^*, a contradiction. □

Corollary 2.2. $p_f < 1$.

Proof: If $p_f \geq 1$, then we have $p_1 \leq w - p_s - 1 < 2 - 1/m - p_s$. Let A_j^* be the subset in the optimal partition σ^* that contains p_s, then the load of A_j^* is

$$C_j^* \leq 2p_1 + p_s < 2\left(2 - \frac{1}{m} - p_s\right) + p_s = 4 - \frac{2}{m} - p_s < w^*,$$

a contradiction to the definition of w^*. □

Define Q_1 to be the minimum load of the subsets at the time right after element p_x is assigned to subset A_u. Recall that Q_0 is the minimum load of the subsets before element p_x is assigned to subset A_u. Clearly, $Q_0 \leq Q_1$. By (2.1) and Corollary 2.2, we know that element p_x comes before element p_f, that is, $x < f$. Therefore, $Q_1 \leq p_1 \leq w - p_s - p_f$. Now we turn to consider element p_{x+1}: if it is larger than $w - p_s - Q_1$, then the subset being assigned with this element will have a load exceeding $w - p_s$ at that time. It follows that enlarging it to p_x would not affect its assignment by the MLPT, whatever it is the second or the third element in the subset. After the assignment, we redefine Q_1 as the minimum load at that time and turn to consider element p_{x+2}. Note from the fact that $p_{l_1} + p_f \leq w - p_s$, that the redefined Q_1 still satisfies $Q_1 \leq w - p_s - p_f$. Repeat this *enlarging-redefining* process for elements $p_{x+2}, p_{x+3}, \ldots$, till we meet some element $p_{x'}$ which is smaller than or equal to $w - p_s - Q_1$, with Q_1 being newly defined (note: it might be the case that $x' = x + 1$—the process enlarges nothing).

Since this enlarging-redefining process doesn't enlarge element p_f, it wouldn't enlarge element p_{k_2} either. Moreover, the MLPT still assigns p_{k_2} to A_k and assigns p_f to A_l, and the subset A_k still has a final load w. That is, I remains as a (minimal) counterexample. Let t_0 denote the time that the enlarging-redefining process terminates. Recall that we have $Q_0 \leq Q_1 \leq w - p_s - p_f$. At time t_0, each of the unassigned elements so far is less than or equal to $w - p_s - Q_1$ (called a *small* element) and each of the already assigned elements is larger than or (enlarged to be) equal to p_x (called a *big* element).

Lemma 2.5. *At time t_0, there is some subset A_i which contains only one big element.*

Proof: We first prove that at time t_0, if A_i contains two or three elements, then its load (at time t_0) is greater than Q_1. This is certainly true if subset A_i has three elements, since its load is at least as large as w. If A_i contains only two elements and its load is at most Q_1, then the final load of subset A_i in the MLPT partition σ is at most $w - p_s$, since its third element must be a small element. This contradicts the definition of w.

But notice that at time t_0 there is some subset whose load is equal to Q_1 by definition. It follows that there is only one element in this subset. □

Lemma 2.6. *Any subset in the optimal partition σ^* contains at least two big elements.*

Proof: From $Q_0 \leq Q_1$ and (2.3), we know that if a subset A_j^* contains at most one big element, then its load

$$\begin{aligned}
C_j^* &\leq (w - p_s - p_f) + 2(w - p_s - Q_1) \\
&= 3w - 2Q_1 - 3p_s - p_f \\
&< 9 - \frac{3}{m} - 2\left(\frac{5}{2} - \frac{1}{2m} - \frac{p_s}{2}\right) - 3p_s - p_f \\
&= 4 - \frac{2}{m} - (2p_s + p_f) \\
&< w^*,
\end{aligned}$$

contradicting the definition of w^*. □

In order to get the final contradiction, we weight the elements as follows:

$$W(p_i) = \begin{cases} 1, & \text{if } p_i \text{ is a small element,} \\ 2, & \text{if } p_i \in [p_x, w - p_s - p_x], \\ 3, & \text{if } p_i \in (w - p_s - p_x, w - p_s - p_f]. \end{cases}$$

We extend this weighting function to a set S of elements to be the total weight of elements in S.

Lemma 2.7. *The weight of set A is $W(A) \geq 6m - 1$ according to the MLPT partition σ.*

Proof: For each subset A_i, if the first element p_{i_1} satisfies $p_{i_1} \leq w - p_s - p_x$, then $W(A_i) \leq 6$. Otherwise, suppose $p_{i_1} > w - p_s - p_x$, but notice that the second element of A_i satisfies $p_{i_2} \leq (w - p_s)/2 \leq w - p_s - p_x$ by Lemma 2.3 and the third element must be small, we have $W(A_i) \leq 6$ too. That is, every subset has a weight at most 6. Looking at subset A_l, as p_f is small and it is the second element of subset A_l, we have $W(A_l) \leq 5$. Therefore, $W(A) \leq 6m - 1$. □

Lemma 2.8. *The weight of set A is $W(A) \geq 6m$ according to the optimum partition σ^*.*

Proof: For each subset A_j^*, if its first element $p_{j_1^*}$ is greater than $w - p_s - p_x$, then clearly we have $W(A_j^*) \geq 6$ by Lemma 2.6. If A_j^* contains only two big elements and they both are less than or equal to $w - p_s - p_x$, then by (2.1) and (2.2) its load

$$\begin{aligned} C_j^* &\leq 2(w - p_s - p_x) + (w - p_s - Q_1) \\ &= 3w - 3p_s - p_x - (Q_1 + p_x) \\ &< 9 - \frac{3}{m} - 3p_s - 1 - \left(4 - \frac{1}{m} - p_s\right) \\ &= 4 - \frac{2}{m} - 2p_s \\ &< w^*, \end{aligned}$$

a contradiction to the definition of w^*. If A_i^* contains three big elements, then definitely $W(A_i^*) \geq 6$. Therefore, $W(A) \geq 6m$. □

Lemma 2.7 and Lemma 2.8 tell us that the weight of set A can be neither greater than $6m - 1$ nor less than $6m$. This is impossible. The impossibility shows that for instance I Case 1 cannot happen.

2.2. *Case 2: p_x is the third element in A_u*

In this case, $Q_0 = p_{u_1} + p_{u_2}$. Clearly we have $Q_0 \leq w - p_s$ and hence $p_x > (4 - 1/m - p_s) - (w - p_s) > 1$ by (2.1), and $Q_0 \geq 2(4 - 1/m - p_s)/3$. It follows that

$p_s \leq w - Q_0 < 1 - 1/m$, which in turn followed $Q_0 > 5/2 - 1/2m - p_s/2$. Therefore, inequalities (2.2), (2.3) and (2.4) still hold in this case. It can be checked that Lemma 2.4 is true too, as well as Corollary 2.2.

Likewise, we define Q_1 to be the minimum load of subsets at the time right after element p_x is assigned to subset A_u. From Lemma 2.4, Corollary 2.2 and $p_x > 1$ we derive that $Q_1 \leq w - p_s - p_f$. Now we arrive at a situation just the same as in Case 1. And the same arguments can show that for instance I Case 2 cannot happen either.

2.3. Tightness

The above discussion says that element p_x cannot be the second element, neither the third element of subset A_u. Therefore such a minimal counterexample I doesn't exist. In other words, we have for any instance I, $w/w^* \geq (3m - 1)/(4m - 2)$. The following example shows the tightness of the performance ratio.

Example 2.1. In this instance, the set $A = \{p_1, \ldots, p_{3m-1}\}$, where $p_i = 2m - \lfloor \frac{i+1}{2} \rfloor$ for $i = 1, 2, \ldots, 2m$, and $p_i = m$ for $i = 2m + 1, 2m + 2, \ldots, 3m - 1$.

It is easy to check that for this instance, $w = 3m - 1$ and $w^* = 4m - 2$. This finishes the proof of Theorem 2.1.

3. Performance of the MLPT applying to KERNEL 3-PARTITIONING

In this section, we examine the performance of the MLPT algorithm applying to KERNEL 3-PARTITIONING under the objective of maximizing the minimum load. We prove the following

Theorem 3.1. *The tight performance ratio of the MLPT is* $(2m - 1)/(3m - 2)$.

In Lin et al. (1998), the authors proved that the worst-case performance ratio of the LPT algorithm applying to $P, g_i \| C_{min}$ is $(2m - 1)/(3m - 2)$. The above theorem states that the additional restriction of at most three elements per subset has no influence on the worst-case behavior of the MLPT algorithm.

Proof: The ideas behind the proof are similar to those in the proof of Theorem 2.1. We first introduce the minimal counterexample, then present several lemmas corresponding to those lemmas in Section 2. The proofs of them can be similarly done and thus omitted.

An *m-counterexample* is an instance of KERNEL 3-PARTITIONING in which set A contains m nonnegative KERNELs and $n \leq 2m$ positive ordinary elements to be partitioned into m subsets, and for which $w/w^* < (2m - 1)/(3m - 2)$. (Note that there should be $m \geq 2$.) A *minimal counterexample* is an m-counterexample in the sense that the parameter m is the minimal, that is, no m'-counterexample exists with $m' < m$. Clearly, if the theorem doesn't hold, then there exists a counterexample. The existence of a counterexample implies the existence of a minimal counterexample. Thus, suppose the minimal counterexample is an m-counterexample and we can show that either it isn't a counterexample or there exists an $(m-1)$-counterexample, we will get a contradiction. Indeed we will do this in the following.

Let I denote this minimal counterexample. W.l.o.g., we assume that the elements of A are sorted as $g_1 \geq g_2 \geq \cdots \geq g_m = 0$ and $p_1 \geq p_2 \cdots \geq p_n$. Similarly as in Section 2, we can assume that in the MLPT partitioning process there is some time at which a closed subset has the least load — the MLPT cannot assign the element under consideration to it because it has already three elements. Obviously, the load of this closed subset is exactly w. Furthermore, we may also assume that there are exactly $2m$ ordinary elements in A, and henceforth in each partition of A every subset contains one kernel and two ordinary elements. Without loss of generality, we assume that in both the MLPT partition σ and the optimal partition σ^*, a subset with index i contains the kernel g_i. In the MLPT partition σ, the elements in A_i are g_i, p_{i_1}, and p_{i_2} with indices $i_1 < i_2$. p_{i_1} and p_{i_2} are called the *first* and the *second* element of A_i, respectively. In the optimal partition σ^*, $A_i^* = \{g_i, p_{i_1^*}, p_{i_2^*}\}$ with indices $i_1^* < i_2^*$, and $p_{i_1^*}$ and $p_{i_2^*}$ are called the *first* and the *second* element of A_i, respectively, as well. Moreover, this time we normalize the elements of A in such a way that $w^* = 3 - 2/m$, and thus $w < 2 - 1/m$. It then follows that the makespan of σ is greater than $3 - 1/m$.

Definition 3.1. A subset $A_i = \{g_i, p_{i_1}, p_{i_2}\}$ in σ is dominated by a subset $A_j^* = \{g_j, p_{j_1^*}, p_{j_2^*}$ in σ^*, if $p_{i_r} \leq p_{j_r^*}$ for $r = 1, 2$ and $g_i \leq g_j$.

During the MLPT partitioning process, let A_k denote the first subset which is assigned two ordinary elements and its load is $C_k^H = w$. Let s denote the index of the second element in A_k. Enlarge all ordinary elements of A which are smaller than p_s to p_s.

Lemma 3.1 (*Domination Lemma* (*Lin et al., 1998*))**.** *For any $i \neq k$, there is no subset A_j^* in σ^* that would dominate subset A_i in σ.*

Proof: The proof is similar to that for Lemma 2.1. □

Lemma 3.2. *During the MLPT partitioning process, at the time p_s is assigned to subset A_k, each of the other subsets contains at least one ordinary element.*

Proof: The proof is similar to that for Lemma 2.2. □

Corollary 3.1. $g_1 \leq w - p_s$.

Proof: The proof is similar to that for Corollary 2.1. □

Let f denote the largest index among those m first ordinary elements in subsets in σ, and suppose p_f is assigned to subset A_l by the MLPT, i.e., $f = l_1$, then $g_l = g_1$ and $f < s$. Enlarge all elements p_i's with $s > i > f$ to p_f.

Lemma 3.3. $p_1 \leq w - p_s$, $p_s < 1 - \frac{1}{m}$, $g_1 \leq w - p_s - p_f$, *and* $p_f < 1$.

Proof: The proof of $p_1 \leq w - p_s$ is done by contradiction. Since the load C_m^* of A_m^* satisfies $w^* \leq C_m^* \leq 2p_1$, we have $p_s < \frac{1}{2} \leq 1 - 1/m$ by Corollary 3.1. The proof of $g_1 \leq w - p_s - p_f$ is similar to the proof of Lemma 2.4; and the proof of $p_f < 1$ is similar to the proof of Corollary 2.2. □

Since the makespan of the MLPT partition σ is greater than $3 - 1/m$, we know that there is some subset A_i whose load exceeds $3 - 1/m - p_s$ at the time p_s is assigned to A_k. During the MLPT partitioning process, let A_u denote the first subset with its load exceeding $3 - 1/m - p_s$ and let p_x denote the element that assigning it to A_u makes the load exceed $3 - 1/m - p_s$. We note that element p_x might be the first or the second element in A_u. Now similarly we trace back the partial MLPT partition obtained at the time right after element p_x is assigned to A_u. Let C_u denote the load of A_u at that time, and $Q_0 = C_u - p_x$, then

$$Q_0 + p_x = C_u > 3 - \frac{1}{m} - p_s. \tag{3.1}$$

We distinguish also two cases to get the contradiction.

3.1. Case 1: $Q_0 < p_x$

In this case, p_x is the first element of A_u and $Q_0 = g_u$. Furthermore,

$$\begin{aligned} p_x &> \frac{3}{2} - \frac{1}{2m} - \frac{p_s}{2}, \\ Q_0 &> \left(3 - \frac{1}{m} - p_s\right) - (w - p_s) > 1. \end{aligned} \tag{3.2}$$

It follows that element p_x comes before element p_f by Lemma 3.3. Define Q_1 to be the minimum load of subsets in σ obtained at the time right after p_x is assigned to A_u. We have $Q_0 \leq Q_1 \leq w - p_s - p_f$. By using a same arguments (the *enlarging-redefining* process) as in Subsection 2.1, we conclude that the enlarging-redefining process terminates before the MLPT assigns elements p_f. Say at time t_0, then we still have $Q_1 \leq w - p_s - p_f$ and each of the unassigned elements is less than or equal to $w - p_s - Q_1$ (called a *small* element) and each of the already assigned elements is larger than or equal to p_x (called a *big* element).

Lemma 3.4. *At time t_0, there is some A_i which contains no ordinary element but kernel g_i.*

Proof: The proof is similar to that of Lemma 2.5. □

Lemma 3.5. *Every subset in the optimal partition σ^* contains at least one big element.*

Proof: The proof is similar to that of Lemma 2.6. □

At time t_0, if no subset contains two big elements, then from Lemma 3.4, we know that set A contains at most $m-1$ big elements. This contradicts Lemma 3.5. Thus we conclude that there is some subset which contains two big elements at time t_0. It then follows that $Q_1 \geq p_x$ (note that p_x is an smallest big element). Combining this with the former part of (3.2), we have

$$Q_1 \geq p_x > \frac{3}{2} - \frac{1}{2m} - \frac{p_s}{2}. \tag{3.3}$$

Lemma 3.6. *If kernel $g_i \geq (w - p_s - p_f) - p_x$, then subset A_i^* contains two big elements.*

Proof: Suppose subset A_i^* contains up to one big element, then by Lemma 3.3 and (3.1), we have

$$\begin{aligned} C_i^* &\leq (w - p_s - p_f) - p_x + (w - p_s) + (w - p_s - Q_1) \\ &= 3w - 3p_s - p_f - p_x - Q_1 \\ &< 6 - \frac{3}{m} - \left(3 - \frac{1}{m} - p_s\right) - 3p_s - p_f \\ &< w^*, \end{aligned}$$

a contradiction to the definition of w^*. □

Now we introduce our second *weighting function* $W(\cdot)$ as follows:

$$W(p_i) = \begin{cases} 1, & \text{if } p_i \text{ is a small element,} \\ 2, & \text{if } p_i \text{ is a big element;} \end{cases}$$

$$W(g_j) = \begin{cases} 1, & \text{if } g_j \leq (w - p_s - p_f) - p_x, \\ 2, & \text{if } g_j > (w - p_s - p_f) - p_x. \end{cases}$$

We define the weight $W(S)$ of a set S as the total weight of elements in S.

Lemma 3.7. *The weight of set A is $W(A) \leq 5m - 1$ according to the MLPT partition σ.*

Proof: For any subset A_i in σ, if $g_i \leq (w - p_s - p_f) - p_x$, then clearly $W(A_i) \leq 5$. If $g_i > (w - p_s - p_f) - p_x$, then subset A_i contains at most one big element since $Q_1 \leq w - p_s - p_f < g_i + p_x$. Therefore, $W(A_j) \leq 5$ too. Looking at subset A_l, as p_f is small and it is the first element of A_l, $W(A_j) \leq 4$. It follows that $W(A) \leq 5m - 1$. □

Lemma 3.8. *The weight of set A is $W(A) \geq 5m$ according to the optimal partition σ^*.*

Proof: For any subset A_i^* in σ^*, if it contains two big elements, then we have $W(A_i^*) \geq 5$. If A_i^* contains only one big element, then by Lemma 3.6, there should be $g_i > (w - p_s - p_f) - p_x$. Therefore, $W(A_i^*) \geq 5$. From Lemma 3.5 we derive that every subset in σ^* has a weight at least 5 and thus $W(A) \geq 5m$. □

The odds between Lemmas 3.7 and 3.8 implies that for instance I Case 1 cannot happen.

3.2. *Case 2*: $Q_0 \geq p_x$

In this case, we have $Q_0 > \frac{3}{2} - \frac{1}{2}m - \frac{p_s}{2}$, i.e., (3.3) holds. On the other hand, we also have $p_x > 1$. Therefore element p_x comes before element p_f too (by Lemma 3.3). Define Q_1 similarly, then we have also $Q_0 \leq Q_1 \leq w - p_s - p_f$. We then arrive at a situation the same as in Case 1. Therefore, the same argument applies and for instance I this case cannot happen either.

3.3. *Tightness*

Since p_x can neither be less than Q_0, nor can it be greater than or equal to Q_0, this minimal counterexample doesn't exist. In other words, we have for any instance, $w/w^* \geq (2m-1)/(3m-2)$. The following example shows the tightness of the performance ratio.

Example 3.1. In this instance, set $A = \{g_1, g_2, \ldots, g_m, p_1, p_2, \ldots, p_{2m-1}\}$, where $g_j = m - j$ for $j = 1, 2, \ldots, m$; and $p_i = 2m - i$ for $i = 1, 2, \ldots, m$, and $p_i = m$ for $i = m+1, m+2, \ldots, 2m-1$.

It is easy to check that for this instance, $w = 2m - 1$ and $w^* = 3m - 2$. This completes the proof of Theorem 3.1. □

References

L. Babel, H. Kellerer, and V. Kotov, "The k-partitioning problem," *Mathematical Methods of Operations Research*, vol. 47, pp. 59–82, 1998.

S.Y. Chang and H.-C. Hwang, "The worst-case analysis of the MULTIFIT algorithm for scheduling nonsimultaneous parallel machines," *Discrete Applied Mathematics*, vol. 92, pp. 135–147, 1999.

B. Chen, "Worst-case performance of scheduling heuristics," Ph.D. Thesis, Erasmus University in Rotterdam, The Netherlands, 1994.

S.P. Chen, Y. He, and E.Y. Yao, "Three-partitioning containing kernels: Complexity and heuristic," *Computing*, vol. 57, pp. 255–272, 1996.

J. Csirik, H. Kellerer, and G. Woeginger, "The exact LPT-bound for maximizing the minimum completion time," *Operations Research Letters*, vol. 11, pp. 281–287, 1992.

B.L. Deuermeyer, D.K. Friesen, and M.A. Langston, "Scheduling to maximize the minimum processor finish time in a multiprocessor system," *SIAM Journal on Algebraic and Discrete Methods*, vol. 3, pp. 190–196, 1982.

M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman: San Francisco, 1978.

R.L. Graham, "Bounds on multiprocessing timing anomalies," *SIAM Journal on Applied Mathematics*, vol. 17, pp. 416–429, 1969.

H. Kellerer and V. Kotov, "A $\frac{7}{6}$-approximation algorithm for 3-partitioning and its application to multiprocessor scheduling," *INFOR*, vol. 37, pp. 48–56, 1999.

H. Kellerer and G. Woeginger, "A tight bound for 3-partitioning," *Discrete Applied Mathematics*, vol. 45, pp. 249–259, 1993.

C.Y. Lee, "Parallel machine scheduling with nonsimultaneous machine available time," *Discrete Applied Mathematics*, vol. 30, pp. 53–61, 1991.

C.Y. Lee, Y. He, and G.C. Tang, "A note on parallel machine scheduling with non-simultaneous machine available time," *Discrete Applied Mathematics*, vol. 100, pp. 133–135, 2000.

G.-H. Lin, E.Y. Yao, and Y. He, "Parallel machine scheduling to maximize the minimum load with nonsimultaneous machine available times," *Operations Research Letters*, vol. 22, pp. 75–81, 1998.