JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 23, 641-647 (2007)

Short Paper

Performances of List Scheduling for Set Partition Problems*

BANG YE WU AND HSIU-HUI OU
Department of Computer Science and Information Engineering
Shu-Te University
Kaohsiung, 824 Taiwan
E-mail: bangye@mail.stu.edu.tw

An m-partition of a set is a way to distribute the members into m parts. Given a set of positive numbers, an optimal m-partition problem asks for an m-partition optimizing some objective function. List scheduling is an on-line algorithm that has been widely used in scheduling problems. In this paper, we show the tight bounds on the performances of list scheduling for partition problems with the following objective functions: maximizing the minimum part, maximizing the sum of the smallest k parts, minimizing the sum of the largest k parts, and minimizing the ratio of the largest to the smallest part, in which $1 \leq k < m$.

Keywords: analysis of algorithms, approximation algorithms, list scheduling, partitions, optimization problems

1. INTRODUCTION

An m-partition of a set is a way to distribute the members into m parts (subsets). Given a set of numbers, an optimal m-partition problem asks for an m-partition optimizing some objective function. In this paper, a set is a multi-set, *i.e.*, the members of a set are not necessarily distinct. Motivated by scheduling problems, the min-max m-partition problem of a set of positive numbers has been widely studied. In such a problem, there are some independent jobs to be scheduled on m identical machines. For each of the jobs, we are given a positive number representing the execution time of the job. To complete the whole job as early as possible, we look for an m-partition of the numbers such that the maximum sum of the subset is minimized. The min-max m-partition problem is obviously NP-hard since it includes the PARTITION problem ([SP12] in [1]) as a special case when $m = 2$, in which we are given a set of integers and asked if the set can be partitioned into two subsets of equal sum.

List Scheduling (LS) is an on-line algorithm that has been widely used in scheduling problems. Starting with m empty sets, it distributes the incoming numbers one by one without any knowledge of the successive data. At each iteration, the incoming number is put into the currently smallest subset. It was shown that the maximum part of the m-parti-

---

Received March 10, 2005; revised June 9, 2005; accepted June 20, 2005.
Communicated by Gen-Huey Chen.

* This work was supported in part by the National Science Council of Taiwan, R.O.C., under grant No. 93-2213-E-366-014.

tion delivered by LS is at most $2 - 1/m$ times of the optimal [2]. Probabilistic analyses of LS are also available in the literature [3, 4].

The previous results focus on the upper bound of the maximum part of an LS partition, but there is no report on the minimum part. Motivated by how to partition a set as even as possible, in this paper, we consider the set partition problems with different objective functions and show the following bounds for the LS algorithm.

- Maximizing the minimum part (max-min problem): the tight bound is m.
- Maximizing the sum of the smallest k parts for any $1 \leq k < m$ (max-k-min problem): the tight bound is m/k.
- Minimizing the sum of the largest k parts for any $1 \leq k < m$ (min-k-max problem): the tight bound is $2 - k/m$.
- Minimizing the ratio of the largest to the smallest part (min-ratio problem): the tight bound is $m + 1$.

Apparently all the four problems are NP-hard since the problem of determining if there exists an even partition is NP-complete. The max-min problem is a special case of the max-k-min problem with $k = 1$. For the sake of easy to read, we shall first discuss the max-min problem, and then show the general case. The min-k-max problem is a generalization of the min-max problem, and the bound shown in this paper includes the previous result [2] as a special case with $k = 1$. Finally we show the bound for min-ratio problem by using the result of the max-min problem.

The rest of the paper is organized as follows. In the next section, we first define some notations, and the performance analyses are in section 3. In section 4, we give concluding remarks.

2. PRELIMINARIES

In this paper, a set is a multi-set, and all its members are positive numbers. Since we consider only LS algorithm, it is convenient to treat the input set as a sequence. We shall assume that $m \geq 2$ is the number of subsets to be partitioned and $S = \{s_i\}_{i=1}^{n}$ is the input sequence of positive numbers, in which $n \geq m$. An LS m-partition of S is the one delivered by LS, and a max-min partition is the one with maximized minimum part. Similarly, a max-k-min, a min-k-max, and a min-ratio partition are respectively the optimal solutions of the three problems. For the completeness, LS algorithm is listed in the following.

Algorithm LS
Input: A sequence $S = \{s_i \mid 1 \leq i \leq n\}$ of positive numbers and an integer m.
Output: An m-partition $Q = \{Q_i \mid 1 \leq i \leq m\}$.

```
1. For $i \leftarrow 1$ to $m$ do
2.     $Q_i \leftarrow \emptyset$;
3. For $i \leftarrow 1$ to $n$ do
4.     insert $s_i$ into the set of smallest sum in $Q$;
5. Output $Q$.
```

For a set V of numbers, let $w(V)$ denote the sum of all the elements in V. In the remaining of this paper, we shall assume that $\mathcal{Q} = \{Q_i \mid 1 \le i \le m\}$ is the LS partition. For our convenience, we relabel the subsets such that $w(Q_i) \ge w(Q_{i+1})$ for each $i < m$. Let x_i, $1 \le i \le m$, be the last number put in Q_i. The following property comes directly from the algorithm. It is simple but plays an important role in the performance analysis.

Lemma 1 For any Q_i and Q_j in the LS partition, $w(Q_i) - x_i \le w(Q_j)$. Particularly, $w(Q_i) - x_i \le w(Q_m)$ for any i.

Proof: For otherwise, x_i should be put in the smallest subset. ❑

3. PERFORMANCES OF LIST SCHEDULING

3.1 The Max-Min Problem

We first consider the minimum part of the LS partition. By $C_{\min}^{LS}$, we denote the sum of the minimum part in a LS partition, *i.e.*, $C_{\min}^{LS} = w(Q_m)$. Let $C_{\min}^{*}$ denote the sum of the minimum part in a max-min partition. Suppose that there exists some s_i larger than $C_{\min}^{*}$. Obviously s_i is in a part of singleton in the max-min partition. Otherwise we can move its partner to the minimum part and obtain another partition at least as good as the max-min partition. Therefore, substitute s_i by $C_{\min}^{*}$ will not reduce $C_{\min}^{*}$. As a result, we can assume that there is no element larger than $C_{\min}^{*}$ in this subsection.

Theorem 1 $C_{\min}^{*} \le mC_{\min}^{LS}$, and the bound is tight.

Proof: By the property of LS, we have $w(Q_i) - x_i \le C_{\min}^{LS}$. Therefore,

$$w(S) = w(Q_m) + \sum_{i=1}^{m-1} w(Q_i) \le w(Q_m) + \sum_{i=1}^{m-1} (C_{\min}^{LS} + x_i) = mC_{\min}^{LS} + \sum_{i=1}^{m-1} x_i.$$

Since the minimum part of any partition is never larger than the average, we have

$$C_{\min}^{*} \le w(S)/m \le \left(mC_{\min}^{LS} + \sum_{i=1}^{m-1} x_i \right) / m = C_{\min}^{LS} + \frac{1}{m} \sum_{i=1}^{m-1} x_i.$$

Since $x_i \le C_{\min}^{*}$ for each i, we have

$$C_{\min}^{*} \le C_{\min}^{LS} + \frac{m-1}{m} C_{\min}^{*}$$

and we obtain that $C_{\min}^{*} \le mC_{\min}^{LS}$. To show that the bound is tight, consider an input sequence $\{s_i\}_{i=1}^{2m-1}$, in which $s_i = 1$ for $1 \le i \le m$ and $s_i = m$ for $m + 1 \le i \le 2m - 1$. For this instance, $C_{\min}^{LS} = 1$ while $C_{\min}^{*} = m$. ❑

3.2 The Max-*k*-Min Problem

In this subsection, we consider the sum of the smallest k parts of the LS partition. Let W_k^L denote the sum of the smallest k parts in the m-partition delivered by LS, and W_k^* denote the maximized sum of the smallest k parts.

Lemma 2 For any m-partition of S and any $1 \le k \le m$, the sum of the largest k parts is at least the sum of the largest k numbers in S.

Proof: Let $\mathcal{P} = \{P_i \mid 1 \le i \le m\}$ be an m-partition of S, in which $w(P_i) \ge w(P_{i+1})$. Let y_i be the ith largest number in S. Apparently $w(P_1) \ge y_1$. Suppose by contradiction that there exists some $h < m$ such that $\sum_{i<h} w(P_i) \ge \sum_{i<h} y_i$ but $\sum_{i\le h} w(P_i) < \sum_{i\le h} y_i$. It is implied that $w(P_h) < y_h$. Since $\sum_{i\le h} w(P_i) < \sum_{i\le h} y_i$, there must exist some $j \le h$ such that $y_j \notin \cup_{i\le h} P_i$. Suppose that $y_j \in P_t$, $t > h$. However, since all numbers are positive, $w(P_t) \ge y_j > w(P_h)$, a contradiction. ❑

The following result can be obtained directly from Lemma 2 and the proof is omitted.

Corollary 1 For any $k < m$, if $S_1 \subset S$ and $|S_1| = m - k$, then $W_k^* \le w(S) - w(S_1)$.

Theorem 2 For any $k \le m$, $W_k^* \le (m/k)W_k^L$, and the bound is tight.

Proof: By the property of LS, $w(Q_i) - x_i \le w(Q_m)$ for any $1 \le i \le m - k$.

$$\begin{aligned} w(S) &= \sum_i w(Q_i) = \sum_{i=1}^{m-k} w(Q_i) + \sum_{i=m-k+1}^{m} w(Q_i) \\ &\le \sum_{i=1}^{m-k} (x_i + w(Q_m)) + W_k^L = (m-k)w(Q_m) + \sum_{i=1}^{m-k} x_i + W_k^L. \end{aligned} \tag{1}$$

By Corollary 1, we have

$$W_k^* \le w(S) - \sum_{i=1}^{m-k} x_i. \tag{2}$$

By Eqs. (1) and (2), we obtain

$$W_k^* \le W_k^L + (m-k)w(Q_m).$$

Since $w(Q_m) \le W_k^* / k$,

$$W_k^* \le W_k^L + (m-k)w(Q_m) \le W_k^L + (m-k)W_k^L / k = (m/k)W_k^L.$$

To show tightness of the bound, consider an input sequence $\{s_i\}_{i=1}^{2m-k}$, in which s_i = 1 for $1 \le i \le m$ and $s_i = m$ for $i > m$. For this instance, $W_k^L = k$ while $W_k^* = m$. ❑

3.3 The Min-*k*-Max Problem

In this subsection, we consider the sum of the largest k parts of the LS partition. Let M_k^L denote the sum of the largest k parts in the LS partition, and M_k^* denote the minimized sum of the largest k parts. By definition, $M_k^L = \sum_{i \le k} w(Q_i)$.

Theorem 3 $M_k^L \le (2 - k/m)M_k^*$, and the bound is tight.

Proof: For any i, by the property of LS, $w(Q_m) \ge w(Q_i) - x_i$. Summing up for i from 1 to k, we have

$$w(Q_m) \ge \frac{1}{k}\sum_{i=1}^{k}(w(Q_i) - x_i) = \left(M_k^L - \sum_{i=1}^{k} x_i\right)/k.$$

Since Q_m is the smallest part,

$$\sum_{i=k+1}^{m} w(Q_i) \ge (m-k)w(Q_m) \ge \frac{m-k}{k}\left(M_k^L - \sum_{i=1}^{k} x_i\right).$$

Then,

$$w(S) = \sum_{i=1}^{k} w(Q_i) + \sum_{i=k+1}^{m} w(Q_i) \ge M_k^L + \frac{m-k}{k}\left(M_k^L - \sum_{i=1}^{k} x_i\right) = \frac{m}{k} M_k^L - \frac{m-k}{m}\sum_{i=1}^{k} x_i.$$

Since $M_k^* \ge (k/m)w(S)$,

$$M_k^* \ge M_k^L - \frac{m-k}{m}\sum_{i=1}^{k} x_i.$$

By Lemma 2, $M_k^* \ge \sum_{i=1}^{k} x_i$, and we have that $M_k^* \ge M_k^L - ((m-k)/m)M_k^*$, which implies $M_k^L \le (2 - k/m)M_k^*$.

To show the bound is tight, consider an input sequence $\{s_i\}_{i=1}^{m(m-k)+k}$, in which $s_i = 1$ for $1 \le i \le m(m-k)$ and $s_i = m$ for $i > m(m-k)$. For this instance, $w(Q_i) = 2m - k$ for $i \le k$ and $w(Q_i) = m - k$ for $i > k$. It is easy to see that there exists a partition in which the sum of each part is equal to m. Therefore $M_k^L = (2m-k)k$ and $M_k^* = mk$. It completes the proof. ❑

3.4 The Min-Ratio Problem

Now we analyze the ratio of the largest to the smallest part of the LS partition. Let $\gamma^{LS} = w(Q_1)/w(Q_m)$ and γ^* denote the minimized ratio of the largest to the smallest part of any m-partition of S. We show the following result.

Theorem 4 $\gamma^{LS} \le (m+1)\gamma^*$, and the bound is tight.

Proof: Remember that x_1 is the last number put into Q_1 and $C_{\min}^{LS} = w(Q_m)$. Since $w(Q_1) \le w(Q_m) + x_1$,

$$\gamma^{LS} = \frac{w(Q_1)}{w(Q_m)} \le \frac{C_{\min}^{LS} + x_1}{C_{\min}^{LS}}.$$

We consider two cases. First, if $x_1 \le mC_{\min}^{LS}$, we have

$$\gamma^{LS} \le m + 1,$$

and the result follows that $\gamma^* \ge 1$. Otherwise, $x_1 > mC_{\min}^{LS}$. By Theorem 1, $C_{\min}^* \le mC_{\min}^{LS}$. By definition, $C_{\min}^*$ is the maximized minimum part among all possible m-partition. Therefore it is an upper bound of the smallest part of the min-ratio partition. In the other hand, since the largest part of any partition cannot be smaller than any element in the set, the largest part of the min-ratio partition is at least x_1. Consequently,

$$\gamma^* \ge \frac{x_1}{C_{\min}^*} \ge \frac{x_1}{mC_{\min}^{LS}},$$

and therefore,

$$\gamma^{LS} / \gamma^* \le \left(\frac{C_{\min}^{LS} + x_1}{C_{\min}^{LS}}\right) / \left(\frac{x_1}{mC_{\min}^{LS}}\right) = m\left(\frac{C_{\min}^{LS} + x_1}{x_1}\right) = m\left(\frac{C_{\min}^{LS}}{x_1} + 1\right) < m + 1.$$

To show that the bound is tight, consider an input sequences $\{s_i\}_{i=1}^{2m-1}$, in which $s_i = 1$ for $1 \le i \le m$ and $s_i = m$ for $m + 1 \le i \le 2m - 1$. For this instance, $\gamma^{LS} = m + 1$ while $\gamma^* = 1$. It completes the proof. ❑

4. CONCLUDING REMARKS

In this paper, we analyze the worst-case performances of list scheduling for several set partition problems. We show that the tight bound for each of the problems.

The LPT (longest processing time) algorithm, a modification of LS, is another famous algorithm for set partition. It first sorts the numbers in decreasing order, and then employs the LS algorithm to find the partition. The LPT is not an on-line algorithm but has a better tight bound of 4/3 − 1/(3m) for the min-max problem [5]. It is an interesting future work to analyze the LPT for the objective functions defined in this paper.

Another important objective function for partition problem is the L_2 metric. There are several reports on the L_2 performance of the LPT algorithm [6-8]. However, to our best knowledge, there is no such report on the LS algorithm, and it is also one of our research directions.

REFERENCES

1. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory*

of NP-Completeness, W. H. Freeman and Company, San Francisco, 1979.
2. R. L. Graham, "Bounds for certain multiprocessing anomalies," *Bell System Technical Journal*, Vol. 45, 1996, pp. 1563-1581.
3. J. Bruno and P. Downey, "Probabilistic bounds on the performance of list scheduling," *SIAM Journal on Computing*, Vol. 15, 1986, pp. 409-417.
4. E. G. Coffman, Jr. and E. N. Gilbert, "On the expected relative performance of list scheduling," *Operations Research Journal*, Vol. 33, 1985, pp. 548-561.
5. R. L. Graham, "Bounds on multiprocessing timing anomalies," *SIAM Journal on Applied Mathematics*, Vol. 17, 1969, pp. 263-269.
6. S. Sahni, "Algorithms for scheduling independent tasks," *Journal of the ACM*, Vol. 23, 1976, pp. 116-127.
7. D. S. Hochbaum and D. B. Shmoys, "Using dual approximation algorithms for scheduling problems: practical and theoretical results," *Journal of the ACM*, Vol. 34, 1987, pp. 144-162.
8. A. K. Chandra and C. K. Wong, "Worst-case analysis of a placement algorithm related to storage allocation," *SIAM Journal on Computing*, Vol. 4, 1975, pp. 249-263.
9. J. Y. T. Leung and W. D. Wei, "Tighter bounds on a heuristic for a partition problem," *Information Processing Letters*, Vol. 56, 1995, pp. 51-57.
10. R. R. Goldberg and J. Shapiro, "A tight bound for the *k*-partition problem on ideal sets," *Operations Research Letters*, Vol. 24, 1999, pp. 165-173.

Bang Ye Wu (吳邦一) was born in Kaohsiung, Taiwan. He received the B.S. degree in 1986 from the Department of Electrical Engineering at Chung Cheng Institute of Technology. He received the M.S. and the Ph.D. degrees in Computer Science from National Tsing Hua University in 1991 and 1999 respectively. During 1986 to 1989, 1991 to 1995, and 1999 to 2000, he worked at Chung-Shan Institute of Science and Technology. Since 2000, he is presently an associate professor and the head of the Department of Computer Science and Information Engineering at Shu-Te University. His research interests include algorithms, graph theory and computational biology.

Hsiu-Hui Ou (歐秀慧) was born in Kaohsiung, Taiwan. She received the B.S. degree in 1991 from the Department of Applied Mathematics at Providence University. She is currently working toward her M.S. degree in the Department of Computer Science and Information Engineering, Shu-Te University, Taiwan. The preliminary version of this paper has been presented at the 22nd Workshop on Combinatorial Mathematics and Computer Theory, sponsored by National Cheng Kung University and held at Tainan, Taiwan in 2005.