Communications and Network, 2013, 5, 90-95
doi:10.4236/cn.2013.51B021 Published Online February 2013 (http://www.scirp.org/journal/cn)

Approximation Schemes for the 3-Partitioning Problems

Jianbo Li[1], Honglin Ding[2]
[1]School of Management and Economics, Kunming University of Science and Technology, Kunming, P. R. China
[2]Department of Mathematics, Yunnan University, Kunming, P. R. China
Email: dinghonglinyn@126.com

Received 2012

ABSTRACT

The 3-partitioning problem is to decide whether a given multiset of nonnegative integers can be partitioned into triples that all have the same sum. It is considerably used to prove the strong NP-hardness of many scheduling problems. In this paper, we consider four optimization versions of the 3-partitioning problem, and then present four polynomial time approximation schemes for these problems.

Keywords: 3-partitioning Problem; Approximation Scheme

1. Introduction

The 3-partitioning problem is a classic NP-complete problem in Operations Research and theoretical computer science [10]. The problem is to decide whether a given multi set of nonnegative integers can be partitioned into triples that all have the same sum. More precisely, for a given multi set S of 3 m positive integers, can S be partitioned into m subsets $S_1, S_2, \cdots, S_m$ such that each subset contains exactly three elements and the sums of elements in the subsets (also called loads or lengths) are equal?

For the optimal versions of the 3-partitioning problem, the following four problems have been considered.

Problem 1[13], [14] MIN-MAX 3-PARTITIONING:

Given a multi set $S = \{p_1, p_2, \cdots, p_{3m}\}$ of 3m nonnegative integers, partitioned *S* into *m* subsets $S_1, S_2, \cdots, S_m$ such that each subset contains exactly three elements and the maximum load of the m subsets is minimized.

Problem 2 [6] MIN-MAX KERNEL 3- PARTITIONING:

Given a multi set $S = \{r_1, r_2, \cdots, r_m; p_1, p_2, \cdots, p_{2m}\}$ of 3m nonnegative integers, where each r_j is a kernel andeach p_j is an ordinary element, partitioned *S* into m subsets $S_1, S_2, \cdots, S_m$ such that (1) each subset contains exactly one kernel, (2) each subset contains exactly three elements, and (3) the maximum load of the m subsets is minimized.

Problem 3 [5] MAX-MIN 3-PARTITIONING:

Given a multi set $S = \{p_1, p_2, \cdots, p_{3m}\}$ of 3m nonnegative integers, partitioned *S* into *m* subsets $S_1, S_2, \cdots, S_m$ such that each subset contains exactly three elements and the minimum load of the m subsets is maximized.

Problem 4 [5] MAX-MIN KERNEL3-PARTITIONING:

Given a multi set $S = \{r_1, r_2, \cdots, r_m; p_1, p_2, \cdots, p_{2m}\}$ of 3m nonnegative integers, where each r_j is a kernel andeach p_j is an ordinary element, partitioned *S* into *m* subsets $S_1, S_2, \cdots, S_m$ such that (1) each subset contains exactly one kernel, (2) each subset contains exactly three elements, and (3) the minimum load of the m subsets is maximized.

The 3-partitioning problems have many applications in multiprocessor scheduling, aircraft maintenance scheduling, flexible manufacturing systems and VLSI chip design (see [3, 13]). Kellerer and Woeginger [14] proposed a Modified Longest Processing Time (MLPT, for short) with performance ratio $4/3 - 1/3m$ for MIN-MAX 3-PARTITIONING. Later, Kellerer and Kotov [13] designed a $7/6$ -approximation algorithm which is the best known result for MIN-MAX 3-PARTITIONING. Chen et al. [6] considered-MIN-MAX KERNEL 3-PAR- TITIONING and proved that MLPT has a tight approximation ratio $3/2 - 1/2m$ .Chen et al. [5] considered MAX-MIN 3-PARTITIONINGand MAX-MIN KERNEL 3-PARTITIONING, and showed that MLPT algorithm has worst performance ratios $(3m-1)(4m-2)$ and $(2m-1)(3m-2)$, respectively. To the best of our knowledge, these are the best results.

A generalization of the 3-partitioning problem is the k-partitioning problem in which km elements have to be partitioned into m subsets each of which contains k elements. For the min-max objective, Babel, et al. [2] showed the relationship between the scheduling problems and the k-partitioning problem, and devised a $4/3$ -approximation algorithm. Upper (lower) bounds and heuristic algorithms for the min-max k-partitioning problem can be found in [7-9]. He et al. [11] investigated the max-min k-partitioning problem and presented an algorithm with performance ratio $\max\{2/k, 1/m\}$. Recently, Bruglieri et al. [4] gave an annotated bibliography of the cardinality constrained optimization problems which contains the k-partitioning problems.

Apparently, all four 3-partitioning problems considered in the current paper are NP-hard in the strong sense. Thus we are interested in designing some approximation algorithms. Recall that a polynomial-time approximation scheme (PTAS) for a minimization problem is a family of polynomial algorithms over all $\varepsilon > 0$ such that for every instance of the problem, the corresponding algorithm produces a solution whose value is at most $(1+\varepsilon)OPT$. Similarly, A PTAS for a maximization problem is a family of polynomial algorithm sover all $\varepsilon > 0$ such that for every instance of the problem, the corresponding algorithm produces a solution whose value is at least $(1-\varepsilon)$ OPT. Since four 3-partitioning problems are NP-hard in the strong sense, designing some PTASs for these problems is best possible.

Note that 3-partitioning problems are closely related to the parallel scheduling problem of minimizing the makes pan in which n jobs have to be assigned to m machines such that the maximum machine load is minimized. Hochbaum and Shmoys [12] first presented a PTAS for the makes pan problem by using dual approximation algorithms. Alon et al. [1] designed some linear time approximation schemes for the parallel machine scheduling problems by using a novel idea of clustering the small jobs into blocks of jobs of small but non-negligible size. The basic strategy of designing PTAS in [1,12] is to construct a new instance with a constant number of different sizes from the original instance, to solve the new instance optimally, and then re-construct a near optimal schedule for the original instance. Note that the approximation schemes in [1, 12] cannot be applied directly to the 3-par- titioning problems, because of the cardinality constraint.

To the best of our knowledge, there are no PTASs for the four 3-partitioning problems. In this paper, we first present four polynomial-time approximation schemes for the3-partitioning problems, respectively. As we shall see later, our result are adaptations of the framework of approximation scheme in [1], but with a new rounding method.

2. The Min-Max Objectives

2.1. Min-max 3-partitioningvv

For a given instance I_1 of MIN-MAX 3-PARTITIONING, we first compute a partition with value L_1 using MLPT algorithm in [14]. Kellerer and Woeginger [14] have proved that $OPT_1 \le L_1 \le 4/3\, OPT_1$, where

OPT_1 denotes the value of the optimal solution for instance I_1.

Let $\lambda_1 = \frac{4}{\varepsilon}$. For any $T \subseteq S$, let $p(T) = \sum_{p_j \in T} p_j$ be the length of set *T*. For each element $p_j \in S$, we round it

up to $p_j' = \frac{p_j}{L_1/\lambda_1}\frac{L_1}{\lambda_1}$, and then we get a new instance I_1' with mult set S'. The following lemma about the relationship between instance I_1 and instance I_1' is important to our approximation scheme.

Lemma 1. The optimal value of instance I_1' is no more than $OPT_1 + \frac{3}{\lambda_1}L_1$.

Note that no element in instance I_1 is more than L_1 by the definition of L_1, and in instance I_1', all elements are integer multiples of $\frac{L_1}{\lambda_1}$. Thus, the number of different elements is atmost $\lambda_1 + 1$ in instance I_1'. Let $n_i^{(1)}$ $(i = 0,1,\ldots,\lambda_1)$ denote the number of elements with size $i\frac{L_1}{\lambda_1}$. Clearly, $\sum_{i=0}^{\lambda_1} n_i^{(1)} = 3m$ .By the fact $\mathrm{OPT}_1 \le L_1$ and Lemma 1, we can conclude that the optimal value of instance I_1' is at most $\left(1+\frac{3}{\lambda_1}\right)L_1$. Define a configuretion C_j as a subset of elements which contains exactly three elements in S' and has length no more than

$$\left(1+\frac{3}{\lambda_1}\right)L_1.$$

It is easy to verify that the number of different configurations is at most $K_1 = (\lambda_1 + 1)^3$, which is a constant. Let a_{ij} denote the number of elements of size $i\frac{L_1}{\lambda_1}$ in configuration C_j and x_j be the variable indicating the number of occurrences of configuration C_j in a solution.

For each $t \in \{1,2,\ldots,\lambda_1 + 3\}$, we construct an integer-linear program ILP_t with arbitrary objective, and that the constraints are:

$$\sum_{j=1}^{K_1} a_{ij}x_j = n_i^{(1)}; i = 0,1,2,\ldots,\lambda_1 \tag{1}$$

$$\sum_{j=1}^{K_1} x_j = m; \tag{2}$$

$$x_j = 0; if\ p(C_j) > t\frac{L_1}{\lambda_1} \quad (3)$$

$$x_j \geq 0; j = 1,2,\ldots,K_1 \quad (4)$$

Here, the constraints (1) and (2) guarantee that each-element is exactly in one subset. The constraints (3) mean that we only use the configuration with length no more $t\frac{L_1}{\lambda_1}$. Obviously,

$$\mathrm{OPT}_1^{'} = \min\min\{t\frac{L_1}{\lambda_1} | ILP_t \text{ hasa feasible solution}\},$$

where $\mathrm{OPT}_1^{'}$ denotes the optimal value of instance $I_1^{'}$. In ILP_t, the number of variables is at most $K_1 = (\lambda_1+1)^3$, and the number of constraints is at most $\lambda_1 + 2 + (\lambda_1+1)^3$. Both are constants, as λ_1 is a constant. By utilizing Lenstra's algorithm in [15] whose running time is exponential in the dimension of the program but polynomial in the logarithms of the coefficients, we can decide whether the integer linear programming ILP_t has a feasible solution in time $O(m)$, where the hidden constant depends exponentially on λ_1. By solving at most K_1 integer linear programs, we get an optimal solution for instance $I_1^{'}$. Since computing L_1 can be done in time $O(mlogm)$ [14], and constructing the integer linear programs can be done in time $O(m)$, we arrive at the following lemma.

Lemma 2.An optimal solution for instance $I_1^{'}$ of MIN-MAX3-PARTITIONING can be computed in time $O(mlogm)$.

For an optimal solution $(S_1^{'}, S_2^{'}, \cdots, S_m^{'})$ for instance $I_1^{'}$, replace each element $p_j^{'} \in S_i{'}$ by element p_j in instance I_1, and then we get a partition $(S_1, S_2, \cdots, S_m)$ for instance I_1. This will not increase the objective. By Lemma 1, we have

$$\max_i \max_i p(S_i) \leq OPT_1 + \frac{3}{\lambda_1}L_1$$
$$\leq \left(1+\frac{4}{\lambda_1}\right)OPT_1 \leq (1+\varepsilon)OPT_1,$$

as $L_1 \leq \frac{4}{3}OPT_1$ and $\lambda_1 = \frac{4}{\varepsilon} \geq \frac{4}{\varepsilon}$. Thus, $(S_1, S_2, \cdots, S_m)$ is a $(1+\varepsilon)$-approximation solution for instance I_1. Hence, we achieve the following theorem.

Theorem 3. There exists a PTAS with running time $O(mlogm)$ for MIN-MAX 3-PARTITIONING.

2.2. Min-max Kernel 3-Partitioning

For a given instance I_2 of MIN-MAX KERNEL 3-PAR-TITIONING, we first compute the value L_2 of the feasible solution produced by the algorithm in [6]. We have $OPT_2 \leq L_2 \leq \frac{3}{2}OPT_2$, where OPT_2 denotes the value of the optimal solution for instance I_2.

Let $\lambda_2 = \frac{9}{2\varepsilon}$. For each element in I_2, we round it up to the next integer multiple of L_2/λ_2 kg, i.e.,

$$r_j^{'} = \frac{r_j}{L_2/\lambda_2}\frac{L_2}{\lambda_2}(j = 1,2,\ldots,m)$$

and $$p_j^{'} = \frac{p_j}{L_2/\lambda_2}\frac{L_2}{\lambda_2}(j = 1,2,\ldots,2m).$$

Then we get a new instance $I_2^{'}$ with multi set $S^{'}$.

Similar to Lemma 1, we can obtain the following lemma.

Lemma 4. The optimal value of instance $I_2^{'}$ is no more than $OPT_2 + \frac{3}{\lambda_2}L_2$.

For convenience, let $R^{'} = \{r_1^{'}, r_2^{'}, \ldots, r_m^{'}\}$. Note that the numbers of different elements in $R^{'}$ and $S^{'} - R^{'}$ are at most $\lambda_2 + 1$ in instance $I_2^{'}$. Let $n_i^{(2)} (i = 0,1,\ldots,\lambda_2)$ and $q_i^{(2)} (i = 0,1,\ldots,\lambda_2)$ denote the number of elements in $R^{'}$ and $S^{'} - R$ with size $i\frac{L_2}{\lambda_2}$, respectively. Clearly, $\sum_{i=0}^{\lambda_1} n_i^{(2)} = m$ and $\sum_{i=0}^{\lambda_1} q_i^{(2)} = 2m$. Define a configuration C_j as a subset of elements, which contains exactly one element in $R^{'}$ and two elements in $S^{'} - R^{'}$ and has length no more than $\left(1+\frac{3}{\lambda_2}\right)L_2$. It is easy to see that the number of different configurations is at most $K_2 = (\lambda_2+1)^2$, which is a constant. Let a_{ij} denote the number of elements in $R^{'}$ of size $i\frac{L_2}{\lambda_2}$ in configura- t i o n C_j and b_{ij} denote the number of elements in $S^{'} - R^{'}$ of size $i\frac{L_2}{\lambda_2}$ in configuration C_j. Let x_j be the variable indicating the number of occurrences of configuration C_j in a solution.

For each $t \in \{0,1,2,\ldots,\lambda_1+3\}$, we construct an integer linear program ILP_t with arbitrary objective, and that the constraints are:

$$\sum_{j=1}^{K_2} a_{ij}x_j = n_i^{(2)}; i = 0,1,2,\ldots,\lambda_1 \quad (5)$$

$$\sum_{j=1}^{K_2} b_{ij}x_j = q_i^{(2)}; i = 0,1,2,\ldots,\lambda_1 \quad (6)$$

$$\sum_{j=1}^{K_2} x_j = m; \quad (7)$$

$$x_j = 0; if\ p(C_j) > t\frac{L_1}{\lambda_1} \quad (8)$$

$$x_j \geq 0; j = 1, 2, \ldots, K_2 \quad (9)$$

As before, by implementing Lenstra's algorithm in [15] at most K_2 times, we can find an optimal solution for instance $I_2^{'}$.

Lemma 5. An optimal solution to instance $I_2^{'}$ of MINMAXKERNEL 3-PARTITIONING can be computed in time $O(mlogm)$.

For an optimal solution $(S_1^{'}, S_2^{'}, \cdots, S_m^{'})$ for instance $I_2^{'}$ replace each element $r_j^{'} \in S_i^{'}$ and $p_j^{'} \in S_i^{'}$ by element r_j and p_j in instance I_2, respectively. And then we get a partition $(S_1, S_2, \cdots, S_m)$ for instance I_2. This will not increase the objective. By Lemma 4, we have

$$\max_i p(S_i) \leq OPT_2 + \frac{3}{\lambda_2} L_2 \leq \left(1 + \frac{9}{2\lambda_2}\right) OPT_2$$

$$\leq (1+\varepsilon) OPT_2 \max_i \max_i p(S_i) \leq OPT_2 + \frac{3}{\lambda_2} L_2 ,$$

$$\leq \left(1 + \frac{9}{2\lambda_2}\right) OPT_2 \leq (1+\varepsilon) OPT_1$$

as $L_2 \leq \frac{3}{2} OPT_2$ and $\lambda_2 = \frac{9}{2\varepsilon} \geq \frac{9}{2\varepsilon}$.

Thus, $(S_1, S_2, \cdots, S_m)$ is a $(1+\varepsilon)$-approximation solution for instance I_2.

Hence, we achieve the following theorem.

Theorem 6. There exists a PTAS with running time $O(mlog\ m)$ for MIN-MAX Kernel 3-PARTITIONING.

3. The Max-Min Objectives

For a given instance I_3 MAX-MIN 3-PARTITIONING, we first compute a partition with value L_3 using $MLPT$ algorithm in [5]. Chen et al. [5] have proved that $\frac{3}{4} OPT_3 \leq L_3 \leq OPT_3$, where OPT_3 denotes the value of the optimal solution for instance I_3

Lemma 7. If there exists an element

$$p_j \geq \frac{4}{3} L_3 \geq OPT_3 ,$$

then there exists an optimal partition in which element p_j and the two smallest elements are in the same subset.

Proof. Without loss of generality, we may assume that $p_1 \geq p_2 \geq \cdots \geq p_{3m-1} \geq p_{3m}$. If $p_1 \geq \frac{4}{3} L_3$, Let $(S_1^*, S_2^*, \ldots, S_m^*)$ be an optimal partition for instance I_3, where $S_1^* = \{p_1, p_{i_1}, p_{i_2}\}$. Note that $p_1 \geq OPT_3$, $p_{i_1} \geq p_{3m-1}$, and $p_{i_2} \geq p_{3m}$. Interchanging p_{i_1} and p_{3m-1}, p_{i_2} and p_{3m}, respectively, cannot decrease the objective function. Thus, we get a new optimal partition in which p_1 and the two smallest elements are in the same subset.

With the help of Lemma 7, while there exists an element no less than $4/3 L_3$, we delete it and the two smallest elements from S, and then handle a smaller instance. Thus, we may assume without loss of generality that in the end each element is less than $4/3 L_3$.

Lemma 8. In any feasible solution for instance I_3, the maximum load of the subsets is less than that $4L_3$.

Let $\lambda_3 = \frac{3}{\varepsilon}$. For each element $p_j \in S$, we round it down to $p_j^{'} = \frac{p_j}{L_3 / \lambda_3} \frac{L_3}{\lambda_3}$, and then we get a new instance $I_3^{'}$.

Lemma 9. The optimal value of instance $I_3^{'}$ is at least $OPT_3 - \frac{3}{\lambda_3} L_3$.

Note that all the elements in $I_3^{'}$ are integer multiples of $\frac{L_3}{\lambda_3}$. Thus, the number of different elements is at most $\frac{4}{3}\lambda_3$ in instance $I_3^{'}$. Let $n_i^{(3)} \left(i = 0, 1, \ldots, \frac{4}{3}\lambda_3 - 1\right)$ denote the number of elements with size $i\frac{L_3}{\lambda_3}$. Clearly, $\sum_{i=0}^{\frac{4}{3}\lambda_3 - 1} n_i^{(3)} = 3m$. By Lemma 8, the maximum load of any feasible solution for instance $I_3^{'}$ is less than $4L_3$. Define a configuration C_j as a subset of elements which contains exactly three elements in $S^{'}$ and has length less than $4L_3$. The number of different configurations is at most $K_3 = \frac{4}{3}\lambda_3{}^3$, which is a constant. Let a_{ij} denote the number of elements of size $i\frac{L_3}{\lambda_3}$ in configuration C_j and x_j be the variable indicating the number of occurrences of configuration C_j in a solution.

For each $t \in \{0, 1, 2, \ldots, 4\lambda_3\}$, we construct an integer-linear program ILP_t with arbitrary objective, and that the constraints are:

$$\sum_{j=1}^{K_3} a_{ij} x_j = n_i^{(3)}; i = 0, 1, 2, \ldots, 4\lambda_3 \quad (10)$$

$$\sum_{j=1}^{K_3} x_j = m; \quad (11)$$

$$x_j = 0;\ \text{if}\ p(C_j) < t\frac{L_1}{\lambda_1} \quad (12)$$

$$x_j \geq 0; j = 1, 2, \ldots, K_3 \quad (13)$$

Here, the constraints (10) and (11) guarantee that each element is exactly in one subset. The constraints (12) mean that we only use the configuration with length no Less than $t\frac{L_3}{\lambda_3}$. Obviously,

$$\mathrm{OPT}_3^{'} = \min\{t\frac{L_3}{\lambda_3} | ILP_t \text{ has a feasible solution}\},$$

where $\mathrm{OPT}_3^{'}$ denotes the optimal value of instance $I_3^{'}$. As in Section 2, by implementing Lenstra's algorithm in [15] at most K_3 times, we get an optimal solution of instance $I_3^{'}$. Since computing L_3 can be done in $O(mlogm)$ [5] and constructing the integer linear programs can be done in $O(m)$, we arrive at the following lemma.

Lemma 10. An optimal solution for instance $I_3^{'}$ of-MIN-MAX 3-PARTITIONING can be computed in time $O(mlogm)$.

For an optimal solution $(S_1^{'}, S_2^{'}, \cdots, S_m^{'})$ for instance $I_3^{'}$, replace each element $p_j^{'} \in S_i^{'}$ by element p_j in instance I_3, and then we get a partition $(S_1, S_2, \cdots, S_m)$ for instance I_3. This will not decrease the objective value. By Lemma 9, we have

$$\min_i p(S_i) \ge OPT_3 - \frac{3}{\lambda_3} L_3 \ge \left(1 - \frac{3}{\lambda_3}\right) OPT_3, \\ \ge (1-\varepsilon) OPT_3$$

as $L_3 \le OPT_3$ and $\lambda_3 = \frac{3}{\varepsilon} \ge \frac{3}{\varepsilon}$. Thus, $(S_1, S_2, \cdots, S_m)$ is a $(1-\varepsilon)$-approximation solution for instance I_3.

Hence, we achieve the following theorem.

Theorem 11. There exists a PTAS with running time $O(mlogm)$ for MAX-MIN 3-PARTITIONING.

Similarly, we can obtain the following theorem. We omit the proof here.

Theorem 12. There exists a PTAS with running time $O(mlogm)$ for MAX-MIN Kernel 3-PARTITIONING.

4. Conclusions

We have presented some PTASs for four optimization-versions of 3-partitioning problem. It is an interesting open question whether some similar PTAS can be developed for general objectives of 3-partitioning problem as in [1].

5. Acknowledgements

The work is supported by the National Natural Science Foundation of China [No. 61063011] and the Tianyuan Fund for Mathematics of the National Natural Science Foundation of China [No. 11126315].

REFERENCES

[1] N. Alon, Y. Azar, G. J. Woeginger and T. Yadid, "Approximation Schemes for Scheduling on Parallel Machines," *Journal of Scheduling*, Vol. 1, 1998, pp. 55-66. doi:10.1002/(SICI)1099-1425(199806)1:1<55::AID-JOS2>3.0.CO;2-J

[2] L. Babel, H. Kellerer and V. Kotov, "The k-partitioning Problem," *Mathematical Methods of Operations Research*, Vol. 47, 1998, pp. 59-82. doi:10.1007/BF01193837

[3] J. Brimberg, W. J. Hurley and R. E. Wright, "Scheduling Workers in a Constricted Area," *Naval Research Logistics*, Vol. 43, 1996, pp. 143-149. doi:10.1002/(SICI)1520-6750(199602)43:1<143::AID-NAV9>3.0.CO;2-B

[4] M. Bruglieri, M. Ehrgott, H. W. Hamacher and F. Maffioli, "An Annotated Bibliography of Combinatorial Optimization Problems with Fixed Cardinality Constraints," *Discrete Applied Mathematics*, Vol. 154, 2006, pp. 1344-1357. doi:10.1016/j.dam.2005.05.036

[5] S. P. Chen, Y. He and G. H. Lin, "3-partitioning for Maximizing the Minimum Load," *Journal of Combinatorial Optimization*, Vol. 6, 2002, pp. 67-80. doi:10.1023/A:1013370208101

[6] S. P. Chen, Y. He and E. Y. Yao, "Three-partitioning Containing Kernels: Complexity and Heuristic. *Computing*, Vol. 57, 1996, pp. 255-272. doi:10.1007/BF02247409

[7] M. Dell' Amico, M. Iori and S. Martello, "Heuristic Algorithms and Scatter Search for the Cardinality Constrained $P||C_{max}$ Problem," *Journal of Heuristics*, Vol. 10, 2004, pp. 169-204. doi:10.1023/B:HEUR.0000026266.07036.da

[8] M. Dell' Amico, M. Iori, S. Martello and M. Monaci, "Lower Bound and Heuristic Algorithms for the k-partitioning Problem," *European Journal of Operational Research*, Vol. 171, 2006, pp. 725-742. doi:10.1016/j.ejor.2004.09.002

[9] M. Dell' Amico and S. Martello, "Bounds for the Cardinality Constrained $P||C_{max}$ Problem. *Journal of Scheduling*, Vol. 4, 2001, pp. 123-138. doi:10.1002/jos.68

[10] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," *W. H. Freeman, San Francisco*, 1979.

[11] Y. He, Z. Y. Tan, J. Zhu and E. Y. Yao, "k-Partitioning Problems for Maximizing the Minimum Load," *Computers and Mathematics with Applications*, Vol. 46, 2003, pp. 1671-1681. doi:10.1016/S0898-1221(03)90201-X

[12] D. S. Hochbaum and D. B. Shmoys, "Using Dual Approximation Algorithms for Scheduling Problems: Theoretical and Practical Results," *Journal of Association for Computing Machinery*, Vol. 34, 1987, pp. 144-162. doi:10.1145/7531.7535

[13] H. Kellerer and V. Kotov, "A $7/6$-approximation Algorithm for3-partitioning and Its Application to Multiprocessor Scheduling," *INFOR*, Vol. 37, 1999, pp. 48-56.

[14] H. Kellerer and G. Woeginger, "A Tight Bound for 3-partitioning," *Discrete Applied Mathematics*, Vol. 45, 1993, pp. 249-259. doi:10.1016/0166-218X(93)90013-E

[15] H. W. Lenstra, "Integer Programming with a Fixed Number of Variables," *Mathematics of Operations Research*, Vol. 8, 1983, pp. 538-548. doi:10.1287/moor.8.4.538