Three-Partitioning Containing Kernels: Complexity and Heuristic*  
S.-P. Chen, Y. He, and E.-Y. Yao, Hangzhou  

Received August 11, 1995; revised December 19, 1995  

Abstract — Zusammenfassung  

Three-Partitioning Containing Kernels: Complexity and Heuristic. Let $\mathcal{G} = \{g_1, g_2, \ldots, g_m\} \cup \{t_1, t_2, \ldots, t_n\}$ be a list of items with nonnegative weights assigned and $k > 2$ be an integer. The objective is to find an assignment of the items to the bins such that all $g_i$ (called kernels) are assigned to different bins, such that no bin contains more than $k$ items, and such that the maximum weight assigned to any bin becomes minimum. In this paper, we first prove that the problem is NP-complete in the strong sense for any $k \geq 3$. As heuristic for this problem, we use a modified version of the famous LPT-algorithm for multiprocessor scheduling, and we show a worst case bound of $\frac{3}{2} - \frac{1}{2m}$ for $k = 3$.  

AMS Subject Classifications: 90B35, 68M20.  
Key words: $k$-partitioning containing kernels, NP-complete, worst case analysis, LPT-algorithm.  

Drei-Partitionierung mit Kernen: Komplexität und Heuristik. Sei $k \geq 2$ eine natürliche Zahl und $\mathcal{G} = \{g_1, g_2, \ldots, g_m\} \cup \{t_1, t_2, \ldots, t_n\}$ eine Menge von höchstens $km$ nichtnegativen ganzen Zahlen. Gesucht ist eine Partition von $\mathcal{G}$ in $m$ Teilmengen, die jeweils nicht mehr als $k$ Elemente enthalten, so daß alle $g_i$ (Kerne genannt) unterschiedlichen Teilmengen zugeordnet werden und die maximale Summe von Zahlen in einer dieser Teilmengen möglichst klein wird. Wir zeigen zunächst, daß für jedes $k \geq 3$ dieses Problem NP-vollständig im starken Sinne ist. Als Heuristik für dieses Problem benutzen wir eine revidierte Version des bekannten LPT-Algorithmus für das Multiprozessorscheduling-Problem. Für $k = 3$ zeigen wir eine Worst-Case Schranke von $\frac{3}{2} - \frac{1}{2m}$.  

1. Introduction  

Set partitioning problems generally ask for a partition of a given set $\mathcal{G}$ of items with positive weight into a given number $m$ of subsets such that the total weights of the item in the subsets are as nearly equal as possible. Generalized three-partitioning is one of the basic NP-complete problems (see [2]), which can be formulated as follows:  

For a given list $\mathcal{G} = \{t_1, t_2, \ldots, t_n\}$ of $n$ items with positive weights $w(t_1) \geq w(t_2) \geq \cdots \geq w(t_n)$, and a set $\beta$ of $m$ bins $B_1, B_2, \ldots, B_m$ with $n \leq 3m$, we look for an assignment of the items to the $m$ bins such that each bin contains at most 3 items and the makespan $\text{max}_{1 \leq i \leq m} w(B_i)$ is as small as possible. Here $w(B_i) = \sum_{t \in B_i} w(t)$ denotes the weights of a bin in an assignment.  

*This research is supported partly by State Science and Technology Commission and NSF of China.
In [6], Kellerer and Woeginger analyzed a heuristic called MLPT for the above problem. Heuristic MLPT always assigns the largest unassigned item of $\mathcal{S}$ to a bin of $\beta$ with the least current weight. As soon as three items have been put into a bin $B$, we call $B$ blocked and remove it from the list $\beta$. The process is repeated until no item is left. If we denote by $W^*(\mathcal{S}, m)$ the makespan of the MLPT-partition and by $W_0(\mathcal{S}, m)$ the makespan of the OPT-partition, then the below approximation bound is tight (see [3]):

$$\frac{W^*(\mathcal{S}, m)}{W_0(\mathcal{S}, m)} \leq \frac{4}{3} - \frac{1}{3m}.$$ 

In [9], Yao, Li and He proposed to study the problem set partitioning containing kernels, which is a set partitioning problem obeying the following additional restriction:

To the given set $\mathcal{S}$, $m$ additional items (called kernels) are added, which must be assigned to $m$ different bins. This problem is NP-complete because if we set all kernels to zero weight, we get the above set partitioning problem. It is easy to prove that the following algorithm called SLPT has the tight worst case bound $\frac{3}{2} - \frac{1}{2m}$ (see [7, 8, 9]).

Algorithm SLPT:

1. assign the $m$ kernels to $m$ different bins;
2. assign the largest unassigned item to a bin with least current weight until no item is left.

Strongly related is the fundamental problem in scheduling theory: Schedule $n$ independent tasks nonpreemptively on a multiprocessor system, the tasks are all available at time zero, but some machines may not be available at time zero. Here, machine available times are taken as special tasks, and called kernels in the set partitioning problem.

Based on the above ideas, in this paper, we consider the following $k$-partitioning problem containing kernels ($k \geq 2$):

Let $\mathcal{S} = \{g_1, g_2, \ldots, g_m, t_1, t_2, \ldots, t_n\}$ be a list of $n + m$ items, where each kernel $g_i$ ($i = 1, \ldots, m$) has nonnegative weight and all other items (called nonkernels) have positive weights. Moreover, we have a set $\beta$ of $m$ bins $B_1, B_2, \ldots, B_m$ with $n \leq (k-1)m$. We look for an assignment of $\mathcal{S}$ to the $m$ bins satisfying the following three constraints:

1. each bin contains exactly one kernel;
2. each bin contains no more than $k$ items;
3. the makespan $\max_{1 \leq i \leq m} w(B_i)$ is as small as possible.