An Approximation Scheme for Bin Packing with Conflicts *

Klaus Jansen[1]

IDSIA, Corso Elvezia 36, 6900 Lugano, Switzerland,
klaus@idsia.ch

Abstract. In this paper we consider the following bin packing problem with conflicts. Given a set of items $V = \{1, \ldots, n\}$ with sizes $s_1, \ldots, s_n \in (0, 1]$ and a conflict graph $G = (V, E)$, we consider the problem to find a packing for the items into bins of size one such that adjacent items $(j, j') \in E$ are assigned to different bins. The goal is to find an assignment with a minimum number of bins.
This problem is a natural generalization of the classical bin packing problem. We propose an asymptotic approximation scheme for the bin packing problem with conflicts restricted to d-inductive graphs with constant d. This graph class contains trees, grid graphs, planar graphs and graphs with constant treewidth. The algorithm finds an assignment for the items such that the generated number of bins is within a factor of $(1 + \epsilon)$ of optimal, and has a running time polynomial both in n and $\frac{1}{\epsilon}$.

1 Introduction

1.1 Problem Definition

In this paper we consider the following bin packing problem with conflicts. The input I of the problem consists of an undirected graph $G = (V, E)$ with a set of items $V = \{1, \ldots, n\}$ and sizes $s_1, \ldots, s_n$. We assume that each item size is a rational number in the interval $(0, 1]$. The problem is to partition the set V of items into independent sets or bins $U_1, \ldots, U_m$ such that $\sum_{i \in U_j} s_i \leq 1$ for each $1 \leq j \leq m$. The goal is to find a conflict-free packing with a minimum number m of bins. For any instance $I = (G = (V, E), (s_1, \ldots, s_n))$, let $SIZE(I) = \sum_{i=1}^{n} s_i$ denote the total size of the n items, and let $OPT(I)$ denote the minimum number of unit size bins needed to pack all items without conflicts. For graph classes not defined in this paper we refer to [8].

One application of the problem is the assignment of processes to processors. In this case, we have a set of processes (e.g. multi media streams) where some of the processes are not allowed to execute on the same processor. This can be for

* This research was done while the author was associated with the MPI Saarbrücken and was supported partially by the EU ESPRIT LTR Project No. 20244 (ALCOM-IT) and by the Swiss Office Fédéral de l'éducation et de la Science project n 97.0315 titled "Platform".

reason of fault tolerance (not to schedule two replicas of the same process on the same cabinet) or for efficiency purposes (better put two cpu intensive processes on different processors). The problem is how to assign a minimum number of processors for this set of processes. A second application is given by storing versions of the same file or a database. Again, for reason of fault tolerance we would like to keep two replicas / versions of the same file on different file server.

Another problem arises in load balancing the parallel solution of partial differential equations (pde's) by two-dimensional domain decomposition [2, 1]. The domain for the pde's is decomposed into regions where each region corresponds to a subcomputation. The subcomputations are scheduled on processors so that subcomputations corresponding to regions that touch at even one point are not performed simultaneously. Each subcomputation j requires one unit of running time and s_j gives the amount of a given resource (e.g. number of used processors or the used storage). The goal of the problem is to find a schedule with minimum total completion time. In general, the created conflict graphs are nonplanar. But if the maximum number of regions touching at a single point is constant, then the mutual exclusion graph is d-inductive with constant d. Other applications are in constructing school course time tables [17] and scheduling in communication systems [10].

1.2 Results

If E is an empty set, we obtain the classical bin-packing problem. Furthermore, if $\sum_{j \in V} s_j \leq 1$ then we obtain the problem to compute the chromatic number $\chi(G)$ of the conflict graph G. This means that the bin packing problem with conflicts is NP-complete even if $E = \emptyset$ or if $\sum_{j \in V} s_j \leq 1$. We notice that no polynomial time algorithm has an absolute worst case ratio smaller than 1.5 for the bin packing problem, unless $P = NP$. This is obvious since such an algorithm could be used to solve the partition problem [7] in polynomial time. For a survey about the bin packing problem we refer to [4]. The packing problem for an arbitrary undirected graph is harder to approximate, because Feige and Kilian [5] proved that it is hard to approximate the chromatic number to within $\Omega(|V|^{1-\epsilon})$ for any $\epsilon > 0$, unless $NP \subset ZPP$.

In [1, 3] the bin packing problem with conflicts and with unit-sizes ($s_j = \frac{1}{\ell}$ for each item $j \in V$) was studied. Baker and Coffman called this packing problem (with unit sizes) Mutual Exclusion Scheduling (short: MES). In [3] the computational complexity of MES was studied for different graph classes like bipartite graphs, interval graphs and cographs, arbitrary and constant numbers m of bins and constant ℓ. Lonc [16] showed that MES for split graphs can be solved in polynomial time. Baker and Coffman [1] have proved e.g. that forest can be scheduled optimally in polynomial time and have investigated scheduling of planar graphs resulting from a two-dimensional domain decomposition problem. A linear time algorithm was proposed in [14] for MES restricted to graphs with constant treewidth and fixed m. Furthermore, Irani and Leung [10] have studied on-line algorithms for interval and bipartite graphs.