On Lifting Procedures for Packing and Scheduling

Mohamed Haouari[1] - Anis Gharbi[1,2]

[1] Combinatorial Optimization Research Group - ROI, Ecole Polytechnique de Tunisie, BP 743, 2078 La Marsa, Tunisia.

[2] Department of Applied Mathematics, Institut Supérieur d'Informatique, Ariana, Tunisia.

1 Introduction

We investigate several new lower bounds for the one-dimensional bin packing problem and for the identical parallel machine scheduling problem. These bounds are computed through the application of a general *lifting procedure* which aims at systematically tightening previously developed lower bounds. We present the results of an extensive computational study which provides evidence that the proposed lifted bounds dominate the best lower bounds from the literature.

2 The one-dimensional bin packing problem

Given a set J of n items where each item j $(1 \leq j \leq n)$ has a positive integer weight w_j, and a set of identical bins of capacity C, the *Bin Packing Problem* (BPP) is to assign each item to exactly one bin in such a way that the cumulated item weight in each bin does not exceed C and the number of loaded bins is minimal. This problem is $\mathcal{NP}$-hard in the strong sense.

Haouari and Gharbi [3] proved (in a slightly different form within the context of multiprocessor scheduling) that if a feasible solution of a BPP with n items fills up exactly b bins, then there is at least a set of k bins $(1 \leq k \leq b)$ which must contain at least $k \lfloor n/b \rfloor + \min(k, n - \lfloor n/b \rfloor b)$ items. This result offers a practical alternative for improving a given lower bound $L(.)$ in the following way. Given a BPP instance defined on a set J of n items, let $m = L(J)$ and define for each $k = 1, ..., m-1$, $\lambda(k, m, n) = k \lfloor n/m \rfloor + \min(k, n - \lfloor n/m \rfloor m)$. Let $S_n^{m,k} \subseteq J$ denote the set of the $\lambda(k, m, n)$ lightest items in J. If $L(S_n^{m,k}) > k$ for some $k = 1, ..., m-1$, then $m+1$ is a valid lower bound for J. It is worth noting that a further improvement can be obtained if all the subsets of J are considered. However, by virtue of the regularity property, the subset of the l heaviest items clearly dominates any subset of J of cardinality l. Let $S_l^{m,k}$ denote the set of the $\lambda(k, m, l)$ lightest items chosen among the l heaviest ones of J $(1 \leq l \leq n)$. Clearly, if $L(S_l^{m,k}) > k$ for some $k = 1, ..., m-1$ and $l = 1, ..., n$ then $m+1$ is a valid lower bound for J.

Based upon these results, an improved lower bound $\bar{L}(.)$ is equal to the minimal value of m satisfying $L(J) \leq m$ and $L(S_l^{m,k}) \leq k \quad \text{for} \quad k = 1, ..., m-1; l = 1, ..., n$. This procedure has been used for lifting the following lower bounds :

- The trivial bound $L_1(J) = \left\lceil \sum_{j \in J} w_j / C \right\rceil$;
- The bound $L_2(.)$ of Martello and Toth [4]: For each integer $w \in [0, \frac{C}{2}]$, let $J_1(w) = \{j \in J : C - w < w_j\}$, $J_2(w) = \{j \in J : \frac{C}{2} < w_j \leq C - w\}$ and $J_3(w) = \{j \in J : w \leq w_j \leq \frac{C}{2}\}$. Then

$$L_2(J) = \max_{w \in [0, \frac{C}{2}]} \{\max\left(|J_1(w)| + |J_2(w)|, |J_1(w)| + L_1(J_2(w) \cup J_3(w))\right)\}$$

- The class of bounds $L_*^{(p)}$ of Fekete and Schepers [2].

A straightforward computation of $\bar{L}(J)$ (which requires $O(n^3)$ computations of $L(.)$) consists in computing $L(S_l^{m,k})$ for all (k, m, l). However, we prove that the only values of l that have to be considered in the computation of $\bar{L}(.)$ are those such that $l = \alpha m + k$, where $1 \leq \alpha \leq \lfloor (n-k)/m \rfloor$. Also, we show that if for given k, m and α, we have $L(S_{\alpha m+k}^{m,k}) \leq k$, then α has not to be considered for larger values of m. Similarly, if for given k and m, we have $L(S_{\alpha m+k}^{m,k}) \leq k$ for all $1 \leq \alpha \leq \lfloor (n-k)/m \rfloor$, then k has not to be considered for larger values of m. The above results yield only $O(n)$ required computations of $L(.)$.

3 The identical parallel machine scheduling problem

Given a set J of n jobs with corresponding processing times $p_1, p_2, ..., p_n$ and m identical machines ($1 < m < n$ and $p_1 \geq p_2 \geq ... \geq p_n$), the problem is to assign each job to exactly one machine with the objective of minimizing the makespan, or the completion time of the last job. This problem is denoted $P \parallel C_{\max}$.

The $P||C_{\max}$ is related to the BPP in the following way [1]. Given a trial value C of the optimal makespan, consider the decision problem which consists in checking whether the n jobs can be processed on the m machines such that the maximum completion time does not exceed C. A positive answer to this feasibility problem is provided if the minimal number of bins (machines) does not exceed m in the BPP defined for the n items (jobs) with weights p_j and bins' capacity C. Otherwise, we conclude that a valid lower bound for the $P||C_{\max}$ is $C + 1$. Consequently, a class of new lower bounds for $P \parallel C_{\max}$ could be directly derived from the newly developed BPP lifted lower bounds.

Moreover, given a lower bound $L(.)$ for the $P \parallel C_{\max}$, a lifted lower bound $\bar{L}(.)$ could be derived in the following way. Let J_k define an instance that is obtained by considering k machines and the $\lambda(k, m, n)$ jobs with smallest processing times. Clearly, a whole family of lifted lower bounds is derived by considering each P_k $(k = 1, ..., m)$. Let $L(J_k)$ denote the respective value of L computed for the instance J_k. This yields the following valid lower bound $\bar{L}(J) = \max_{1 \leq k \leq m} L(J_k)$. Here again, tighter bounds could be obtained by systematically considering different subsets of J.

In addition to BPP-based lower bounds, this lifting procedure has been applied to strengthen the simple bound $L_1 = \max(\left\lceil \sum_{j \in J} p_j/m \right\rceil, p_m + p_{m+1}, p_1)$ and a newly proposed lower bound, denoted by LB_{SSP}, which is based on the exact solution of subset-sum problems. This bound is

$$LB_{SSP} = \max_{1 \leq k \leq \lfloor m/2 \rfloor} \left(\min(z_1^k, z_2^k) \right)$$

where

$$z_1^k = \min \sum_{j \in J} p_j x_j / k \qquad\qquad z_2^k = \min \sum_{j \in J} p_j (1 - x_j)/(m - k)$$

$$\sum_{j \in J} p_j x_j \geq \left\lceil (k/m) \sum_{j \in J} p_j \right\rceil \qquad\qquad \sum_{j \in J} p_j x_j \leq \left\lfloor (k/m) \sum_{j \in J} p_j \right\rfloor$$

$$x_j \in \{0, 1\}, \ \ j \in J \qquad\qquad x_j \in \{0, 1\}, \ \ j \in J$$

LB_{SSP} can be computed in a pseudo-polynomial time.

4 Computational results

In order to assess the performance of the proposed lifted bounds, we conducted an extensive computational study. For the BPP, we observed that the improved bounds often strictly dominate the original ones. More interestingly, although the bounding procedure of Fekete and Schepers is known to perform remarkably well, the proposed procedure was able to yield even tighter bounds for many instances. Also, even when the original bound is rather poor, the improved one often equals the best one. For instance, we observed that the trivial lower bound L_1 yields a remarkably tight improved lower bound which was found equal to the best one for about 94% of the generated instances. Finally, the improved lower bounds require very short computing times. For $n = 1000$, the largest average computation time was only 0.04 sec. on a Pentium IV 2.8 GHz personal computer. A similar excellent behavior was also observed for $P \,||\, C_{\max}$.

References

[1] M. Dell'Amico and S. Martello (1995). *Optimal Scheduling of tasks on Identical Parallel Processors.* ORSA Journal on Computing 7, 191-200.

[2] S. Fekete and J. Schepers (2001). *New classes of fast lower bounds for bin packing problems.* Mathematical Programming 91, 11-31.

[3] M. Haouari and A. Gharbi (2004). *Lower bounds for scheduling on identical parallel machines with heads and tails.* Annals of Operations Research 129, 187-204.

[4] S. Martello and P. Toth (1990). *Knapsack problems: algorithms and computer implementations.* Wiley, New York.