CS294-1 On-line Computation & Network Algorithms	**Spring 1997**
Lecture 25: May 2, 97	
Lecturer: Yair Bartal	*Scribe: Tina Wong*

This lecture is about on-line scheduling.

25.1 On-line Scheduling and Load Balancing

Problem:

Given a set of M machines, and a sequence of jobs with processing times $p_1, p_2, \ldots, p_j$. Schedule jobs on machines that minimizes the makespan = $\max_{machine\ i} \sum_{j\ sched\ on\ i} p_j$.

Model:

- Identical machines: constant c.r.
- Related machines: also constant c.r. Each machine has speed, $v_1 \leq v_2 \leq \ldots \leq v_m$, and throughput is $\frac{p_j}{v_i}$ makespan = $\max \sum \frac{p_j}{v_i}$
- Unrelated machines: $O(\log m)$ c.r. Each machine has no fixed speed.

25.1.1 Identical machines

Graham analyzed in 1966 the LIST scheduling algorithm for scheduling identical machines [Graham66]. LIST is basically the greedy algorithm: always schedule a job on the least loaded machine.

Theorem 25.1 *The competitive ratio of LIST is* $2 - \frac{1}{m}$.

Proof:

$$
\begin{aligned}
OPT &\geq \max_j p_j \\
OPT &\geq \frac{\sum_j p_j}{m} \\
LIST &= \min_j + p_j \\
&\leq \frac{\sum_{l<j} p_l}{m} + p_j
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\sum_{l \le j} p_l}{m} + (1 - \frac{1}{m}) p_j \\
&\le (2 - \frac{1}{m}) \text{OPT}.
\end{aligned}
$$

■

Some facts:

- $LIST$ is optimal for $m = 2, 3$
- A c.r. $= 2 - \epsilon$, for all m first given by Bartal et al. [BFKV92] and improved in [KPT94, Albers97].
- Currently best bounds are $1,85 \le$ c.r. ≤ 1.92 is due to Albers [Albers97].

25.1.2 Related machines

Aspnes et al. [AAFPW92] prove the following result.

Theorem 25.2 *There exists an 8-competitive algorithm for related machines.*

We note that the greedy algorithm has c.r. logarithmic in the number of machines.

We now prove Theorem 25.2. First, assume that while $L < OPT \le 2L$, we have an algorithm with load $\le 4L < 4 \times OPT$.

If load $> 4L$ then OPT is not $\le 2L$ and we start a new phase with L doubled.

In the final phase,

$$
\begin{aligned}
4L + 4(\frac{1}{2}) + 4(\frac{1}{4}) \ldots &\le 8L \\
&< 8 \times OPT
\end{aligned}
$$

We now turn to deal with the case that the OPT value is indeed bounded as we assumed.

Lemma 25.3 *If $L < OPT \le 2L$, then the algorithm that assigns a job to the slowest machine while ensuring that the load is never $\ge 4L$ never fails.*

Proof: If we can show that everything before the last job can be scheduled within load $2L$, then we are done, because it must be possible to run the last job in at most $2L$ time (since OPT takes $\le 2L$ time). We now extend this idea to jobs other than the last one.

Let the machines have speed $v_1, v_2, \ldots, v_m$. Let S_j be the total size of the jobs that are scheduled on machines which are not in the first j. Also, let $C_j = \sum_{i>j} v_i$. Then we claim that $S_j \leq 2L \cdot C_j$ for all j. To see this, we use induction: there are two cases.

Case I: If $l_j > 2L$, then since $S_{j+1} = S_j - l_j v_j$, we have $S_{j+1} \leq 2L \cdot C_j - 2Lv_j = 2L(C_j - v_j) = 2LC_{j+1}$.

Case II: Let $l_j \leq 2L$. Now $S_{j+1} = \sum p_u$, where the sum is over all jobs u of size greater than $2Lv_j$. Any such job could only be scheduled on machine $j+1$ or later. Therefore the latter sum is bounded above by $2L \cdot C_{j+1}$. ■

References

[Albers97] S. Albers. Better Bounds for On-line Scheduling. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, El-Paso, 1997.

[AAFPW92] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin and O. Waarts. On-line Load Balancing with Applications to Machine Scheduling and Virtual Circuit Routing. In *Proc. of the 33rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 164-173, October 1992.

[BFKV92] Y. Bartal, A. Fiat, H. Karloff, and R. Vohra. New Algorithms for an Ancient Scheduling Problem. In *Proceedings of the 24th ACM Symposium on Theory of Computing*, Victoria, Canada, 1992, 51-58.

[BKR94] Y. Bartal, H. Karloff, and Y. Rabani. A New Lower Bound for m-Machine Scheduling. In *Information Processing Letters*, 50 (1994), 113-116.

[Graham66] R.L. Graham. Bounds for certain multiprocessing anomalies. In *Bell System Technical Journal* 45 (1966), 1563-1581.

[KPT94] D. Karger, S. Phillips, and E. Torng. A Better Algorithm for an Ancient Scheduling Problem. In *Journal of Algorithms*, 20(2): 400-430, 1996.