Online Algorithms	14/12/2011

Lecture Notes 7: Load Balancing and Routing

Professor: Yossi Azar *Scribe: Yigal Lazarev*

1 Introduction

In the first part of this lecture we show a $\lceil \log m \rceil + 1$ competitive algorithm for Restricted-Assignment, and prove it is $2\lceil \log m \rceil + 1$ competitive for simplicity. In the second part we will discuss the generalization heirarchy of the load balancing group of problems, and prove that the Exponent algorithm for the Unrelated model is $O(\log m)$ competitive. The last part of the lecture will present the Routing problem, and show a $O(\log m)$ competitive algorithm for the Generalized Routing problem.

2 Reminder

In the last lecture we saw:

- 8 competitive algorithm for Related Machines model.
- Doubling - if there exists a c competitive algorithm given $\lambda \geq$ OPT, then there exists a $4c$ competitive algorithm without any additional knowledge. The algorithm for RM uses a 2 competitive algorithm given OPT.

3 Restricted Assignment

3.1 Problem Definition

We are given m identical machines, and jobs are arriving on-line one after the other. Job i has weight w_i and a set $\emptyset \neq M(i) \subseteq \{1...m\}$ of machines that can accept it. Job i has to be assigned to a machine $j \in M(i)$.

3.2 Algorithms

Theorem 3.1 *(proved in previous class) Every deterministic algorithm for RA is at least* $\lfloor \log m \rfloor + 1$ *competitive.*

Theorem 3.2 *(proved in previous class) Every randomized algorithm for RA is at least* $\frac{1}{2} \log m$ *competitive.*

Reminder: Greedy algorithm - put job i on machine $j \in M(i)$ with minimal current load.

Theorem 3.3 *Greedy for RA is* $\lceil \log m \rceil + 1$ *competitive.*

Proof We will prove the algorithm is $2\lceil \log m \rceil + 1$ competitive. Denote:

- $\lambda = \mathrm{OPT}(\sigma)$
- For job $i, G(i)$ denotes the machine that received the job in the algorithm.
- $l_j = \sum_{i|G(i)=j} w_i$
- R_i is the volume above height $2\lambda i$. $R_0 = \sum_{j=0}^{m} l_j = \sum_i w_i$.

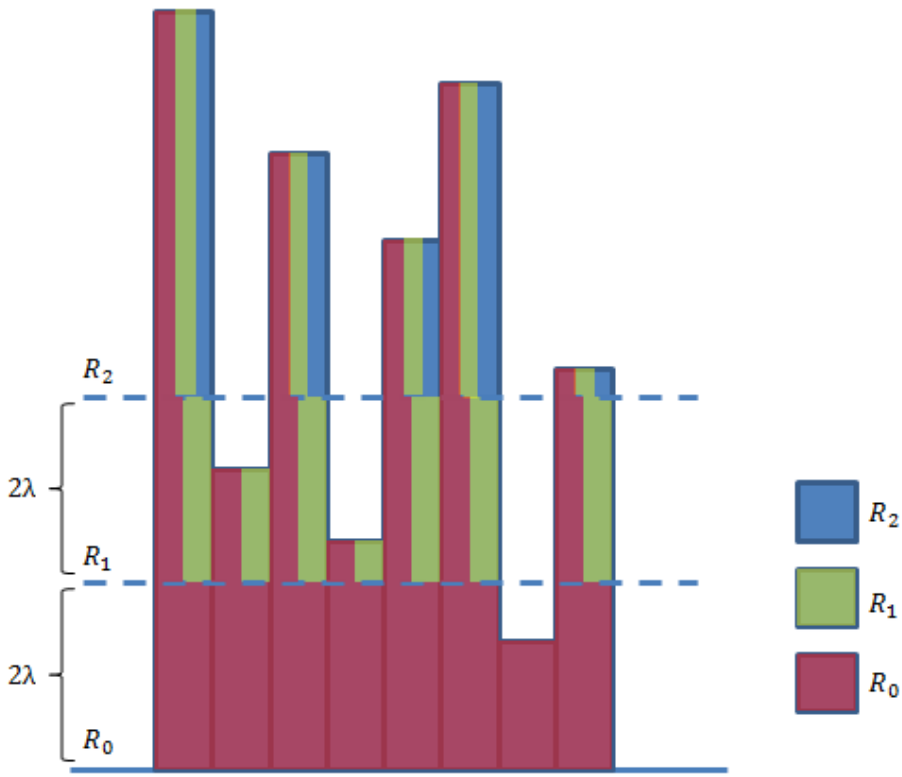

Figure 1: R_i example for $i = 0, 1, 2$

We will show that (I) $R_{i+1} \leq \frac{1}{2} R_i \implies R_i \leq \frac{1}{2^i} R_0 \implies R_{\lceil \log m \rceil} \leq \frac{R_0}{m}$, but $\lambda \geq \frac{R_0}{m} = \frac{\sum_i w_i}{m}$, therefore $\forall j, l_j \leq (2\lceil \log m \rceil + 1)\lambda$. To prove (I), suppose that above height $L \geq 2\lambda$ there is volume $k\lambda$, and show that (II) above height $L - 2\lambda$ there is at least $2k\lambda$ of load, which will conclude the proof. Now all that is left is to prove (II). At height L (and below) OPT must assign the load to at least $\lceil k \rceil$ machines (since every machine is loaded at most λ). We observe the load on these $\lceil k \rceil$ machines. Every job above height L starts at height $\geq L - \lambda$ (because it's size is $\leq \lambda$) therefore each of the $\lceil k \rceil$ machines of OPT is loaded at least $L - \lambda$ (otherwise we would have used it). So between height $L - \lambda$ and $L - 2\lambda$ there is at least $\lceil k \rceil \lambda$ of load, hence above height $L - 2\lambda$ there is at least $\lceil k \rceil \lambda + k\lambda \geq 2k\lambda$ load. ∎

4 Unrelated Machines Model

So far we saw the Identical machines model (with $2-\frac{1}{m}$ competitive algorithm), the Related machines model which generalizes it (with 8 competitive algorithm), and the Restricted Assignment model (with $2\lceil \log m \rceil + 1$ competitive algorithm) which also generalizes the Identical machines model. Restricted Assignment and Related Machines are incomparable. We can generalize the Related and Restricted Assignment models to a Restricted-Related model, in which machine j has speed V_j, and job i has a restriction set $M(i)$. However we can generalize the model even further and define the Unrelated model.

4.1 Problem Definition

We are given m machines, and jobs are arriving on-line. Job i has weight $p_{ij} \in \mathbb{R}^+ \cup \infty$ when assigned to machine j. It is easy to see how this model is a generalization of the Related and Restricted Assignment models. To present a Related machines problem, set $p_{ij} = \frac{w_i}{v_j}$. To present a RA problem, set $p_{ij} = \begin{cases} w_i & j \in M(i) \\ \infty & otherwise \end{cases}$.

Note In the Unrelated model, the volume conservation rule does not apply!

4.2 Algorithms

We will now discuss some possibilities for intuitive and not-so-intuitive algorithms and see their competitive ratio.

- *Post-greedy* Assign on the machine that will have minimal load after assignment. This algorithm is at least m competitive. Example sequence:

$$p_{ij} = \begin{pmatrix} 1+\varepsilon & 1 & & & & \\ & 1+\varepsilon & 2 & & & \\ & & 1+\varepsilon & 3 & & \\ & & & \ddots & & \\ & & & & 1+\varepsilon & m-1 \\ m & & & & & 1+\varepsilon \end{pmatrix}$$

  Note that OPT can assign every job to a machine that will gain $1+\varepsilon$ load after the assignment, so $\mathrm{OPT}(\sigma) \leq 1+\varepsilon$, but PG will choose for job i a machine that will be loaded i, so $\mathrm{PG}(\sigma) = m$, hence at least m competitive.

- Assign job i on machine j such that p_{ij} is minimal. This algorithm is also at least m competitive. Example:

$$p_{ij} = \begin{pmatrix} 1 & & & & & \\ 1 & 1+\varepsilon & & & & \\ 1 & & 1+\varepsilon & & & \\ 1 & & & 1+\varepsilon & & \\ \vdots & & & & \ddots & \\ 1 & & & & & 1+\varepsilon \end{pmatrix}$$

As with Post-Greedy, OPT can assign job i to machine i to get $\mathrm{OPT}(\sigma) \leq 1+\varepsilon$, but the algorithm will always choose machine 1, so $\mathrm{ALG}(\sigma) = m$, hence at least m competitive.

- *Exponent* Suppose $\lambda \geq \mathrm{OPT}$ is known. Put job i on machine j such that

$$a^{\frac{h_j+p_{ij}}{\lambda}} - a^{\frac{h_j}{\lambda}}$$

for some $1 < a < 2$ (to be defined) is minimal, where h_j - the load on machine j. This is equivalent to putting job i on machine j such that after the assignment $\sum_{j=1}^{m} a^{\frac{h_j}{\lambda}}$ is minimal. We divide by λ to get a pure number in the exponent. To shorten the expressions, we normalize the variables and define $\forall x, \tilde{x} = \frac{x}{\lambda}$. We are looking for minimal $a^{\tilde{h_j}+p_{\tilde{i}j}} - a^{\tilde{h_j}}$.

Lemma 4.1

$$\forall a > 1 \textit{and } 0 \leq x \leq 1, a^x - 1 \leq (a-1)x$$

Proof The function $f(x) = (a-1)x$ is linear in the section $[0,1]$, but $g(x) = a^x - 1$ is convex. The functions intersect at $x = 0$ and $x = 1$, and the lemma follows immediately because the functions are continuous. ∎

Theorem 4.2 *The Exponent algorithm is $O(\log m)$ competitive.*

Proof Let us observe job i. Denote h_j^i as the load on machine j before job i arrived, and j^* as the machine OPT decided to assign job i to. From the definition of the algorithm and the lemma, it follows that:

$$a^{\tilde{h_j^i}+p_{\tilde{i}j}} - a^{\tilde{h_j}} \leq a^{\tilde{h_{j^*}^i}+p_{\tilde{i}j^*}} - a^{\tilde{h_{j^*}}} = a^{\tilde{h_{j^*}}}(a^{p_{\tilde{i}j^*}} - 1) \leq a^{\tilde{l_{j^*}}}(a-1)p_{\tilde{i}j^*}$$

where l_{j^*} is the final load. By summing both sides of the equation for all i we get:

$$\sum_i \left(a^{\tilde{h_{j(i)}^i}+p_{i\tilde{j}(i)}} - a^{\tilde{h_{j(i)}^i}} \right) \leq (a-1) \sum_i a^{\tilde{l_{j(i)}^*}} p_{i\tilde{j}_{(i)}^*}$$

By replacing the order of summation we get:

$$\sum_{j=1}^{m} \sum_{i|A(i)=j} \left(a^{\tilde{h_j}+p_{\tilde{i}j}} - a^{\tilde{h_j}} \right) \leq (a-1) \sum_{j^*} a^{\tilde{l_{j^*}}} \sum_{i|\mathrm{OPT}(i)=j^*} p_{\tilde{i}j^*}$$

For each j this is a telescopic sum and hence:

$$\sum_{j=1}^{m} \left(a^{\tilde{l_j}} - a^0 \right) \leq (a-1) \sum_{j=1}^{m} a^{\tilde{l_j}}$$

or

$$\sum_{j=1}^{m} a^{\tilde{l_j}} - m \leq (a-1) \sum_{j=1}^{m} a^{\tilde{l_j}}$$

therefore

$$\sum_{j=1}^{m} a^{\tilde{l_j}} \leq \frac{m}{2-a} \Rightarrow \max_j a^{\tilde{l_j}} \leq \frac{m}{2-a}$$

take log out of both sides to get:

$$\max_j \tilde{l_j} \leq \frac{\log m}{\log a} - \frac{\log(2-a)}{\log a} = O(\log m)$$

∎

5 Routing

5.1 Problem Definition

Given a directed graph $G = (V, E)$ and $C : E \to \mathbb{R}^+$, requests are of the form (s_i, t_i, d_i) and to satisfy them the algorithm needs to choose a path Q_i from s_i to t_i. We define $l_e = \frac{1}{c(e)} \sum_{i|e \in Q_i} d_i$. The goal is to minimize $\max_e l_e$.

5.2 Generalized Routing

Given a directed graph $G = (V, E)$, requests are of the form $\left(s_i, t_i, \{p_{ie}\}_{e \in E}\right)$, where the load on edge e increases by p_{ie} if request i is passing through it. It is easy to see that this is a generalization of the Routing problem if we set $p_{ie} = \frac{d_i}{c(e)}$.

5.3 Algorithm

We will see that the same algorithm as for the Unrelated Scheduling problem (Exponent) works for GR too. WLG $\lambda = 1$ (hence no need to normalize). Denote $w_e = a^{h_e + p_{ie}} - a^{h_e}$. The algorithm will choose a path Q_i such that $\sum_{e|e \in Q_i} w_e$ is minimal.

Theorem 5.1 *Algorithm Exponent is $O(\log m)$ competitive for GR ($m = |E|$).*

Proof For job i, from the algorithm definition we know that:

$$\sum_{e \in Q_i} a^{h_e^i + p_{ie}} - a^{h_e^i} \leq \sum_{e \in Q_i^*} a^{h_e^i + p_{ie}} - a^{h_e^i} = \sum_{e \in Q_i^*} a^{h_e} \left(a^{p_{ie}} - 1\right) \leq \sum_{e \in Q_i^*} a^{l_e} \left(a - 1\right) p_{ie}$$

Summarize both sides for all i:

$$\sum_i \sum_{e \in Q_i} \left(a^{h_e + p_{ie}} - a^{h_e}\right) \leq \sum_i \sum_{e \in Q_i^*} a^{l_e} (a-1) p_{ie}$$

By replacing the order of summation we get:

$$\sum_{e \in E} \sum_{i|e \in Q_i} \left(a^{h_e + p_{ie}} - a^{h_e}\right) \leq (a-1) \sum_{e \in E} a^{l_e} \sum_{i|e \in Q_i^*} p_{ie}$$

For each $e \in E$ we have a telescopic sum:

$$\sum_{e \in E} \left(a^{l_e} - a^0 \right) \leq (a-1) \sum_{e \in E} a^{l_e} \cdot 1$$

or

$$\sum_{e \in E} a^{le} - |E| \leq (a-1) \sum_{e \in E} a^{l_e}$$

hence

$$\sum_{e \in E} a^{le} \leq \frac{|E|}{2-a} \Rightarrow \max_e a^{l_e} \leq \frac{|E|}{2-a}$$

Again take log out of both sides:

$$\max_e l_e \leq \frac{\log |E|}{\log a} - \frac{\log(2-a)}{\log a} = O(\log |E|)$$

$\blacksquare$

Note We did not use the fact that Q_i is a path.