Online Scheduling with Predictions

Thomas Lavastida

Carnegie Mellon University

tlavastida@cmu.edu

May 25, 2018

with Ben Moseley, Silvio Lattanzi and Sergei Vassilvitskii

Load Balancing w/ Unrelated Machines

- m machines
- Sequence of n jobs

Load Balancing w/ Unrelated Machines

- m machines
- Sequence of n jobs
- Job j has processing time $p_{ij} \geq 0$ on machine i

Load Balancing w/ Unrelated Machines

- m machines
- Sequence of n jobs
- Job j has processing time $p_{ij} \geq 0$ on machine i
- Each job must be assigned to a single machine

Load Balancing w/ Unrelated Machines

- m machines
- Sequence of n jobs
- Job j has processing time $p_{ij} \geq 0$ on machine i
- Each job must be assigned to a single machine
- Goal is to minimize the maximum load
- Load of a machine = total processing time of jobs assigned to it

Online Load Balancing w/ Unrelated Machines

- Sequence of jobs arrive *online*
- Uncertainty about future jobs

Online Load Balancing w/ Unrelated Machines

- Sequence of jobs arrive *online*
- Uncertainty about future jobs
- Immediate dispatch - assign jobs upon arrival
- Reassignments not allowed

Online Load Balancing w/ Unrelated Machines

- Sequence of jobs arrive *online*
- Uncertainty about future jobs
- Immediate dispatch - assign jobs upon arrival
- Reassignments not allowed
- Worst case analysis against offline optimal algorithm
- Algorithm A is c-competitive if for all job sequences σ:

$$A(\sigma) \leq c \cdot OPT(\sigma)$$

What is known?

Offline:

- NP-hard
- PTAS for constant m
- 2-approximation for general m
  - Lenstra, Shmoys, and Tardos

What is known?

Offline:

- NP-hard
- PTAS for constant m
- 2-approximation for general m
  - Lenstra, Shmoys, and Tardos

Online:

- $\Omega(\log n)$ lower bound on the competitive ratio for *any* online algorithm
  - Holds for deterministic and randomized online algorithms
  - Azar, Naor, and Rom
- $O(\log n)$-competitive algorithms known
  - Aspnes, et al

Going Beyond Worst Case

- Reassignments
- Stochastic and queueing models
- Retain desirable properties of worst case analysis
- Is there something that is useful to predict?
  - O(1)-competitive online algorithm

Predictions

- Predict the distribution of jobs
  - Number of job types could be large
  - Even with $p_{ij} \in \{1, \infty\}$, there are $O(2^m)$ job types

Predictions

- Predict the distribution of jobs
  - Number of job types could be large
  - Even with $p_{ij} \in \{1, \infty\}$, there are $O(2^m)$ job types
- Can we find a useful prediction that is *compact*?
- Are the predictions *robust* to small errors?
- Captures the contentiousness of a machine
- Ideally lead to $O(1)$-competitive online algorithm

Current Direction

- Write down mathematical program and its dual
  - Constraint for assigning each job
  - Constraint for maintaining the load of each machine

Current Direction

- Write down mathematical program and its dual
  - Constraint for assigning each job
  - Constraint for maintaining the load of each machine
- Predict dual variables for machine constraints
  - Compact prediction
  - Captures cost/contentiousness of a machine
  - Reconstruct primal solution online
  - Robust?

Current Direction

- Write down mathematical program and its dual
  - Constraint for assigning each job
  - Constraint for maintaining the load of each machine
- Predict dual variables for machine constraints
  - Compact prediction
  - Captures cost/contentiousness of a machine
  - Reconstruct primal solution online
  - Robust?
- Round the fractional solution online

Recap

- Online Load Balancing w/ Unrelated machines

Recap

- Online Load Balancing w/ Unrelated machines
  - $\Theta(\log n)$-competitive in worst case model

Recap

- Online Load Balancing w/ Unrelated machines
  - $\Theta(\log n)$-competitive in worst case model
- Want to use predictions to break through this

Recap

- Online Load Balancing w/ Unrelated machines
  - $\Theta(\log n)$-competitive in worst case model
- Want to use predictions to break through this
- Compact - only a few entries per machine

Recap

- Online Load Balancing w/ Unrelated machines
  - $\Theta(\log n)$-competitive in worst case model
- Want to use predictions to break through this
- Compact - only a few entries per machine
- Robust - reconstruct good solution in the presence of small errors

Recap

- Online Load Balancing w/ Unrelated machines
  - $\Theta(\log n)$-competitive in worst case model
- Want to use predictions to break through this
- Compact - only a few entries per machine
- Robust - reconstruct good solution in the presence of small errors
- O(1)-competitive online algorithm

Appendix

Our approach

Assumptions/Notation:

- We know the optimal makespan T^*.
- For each machine i: $\Gamma_i := \{j \mid p_{ij} \leq T^*\}$
- For each job j: $\Gamma_j := \{i \mid p_{ij} < \infty\}$

General Approach:

1. Write down convex programming relaxation
2. Take the dual!
3. Predict dual variables
4. Reconstruct primal solution

Relaxation

Variables: x_{ij} - allocation of job j to machine i
Constraints:

- The load of each machine is at most T^*

$$\sum_{j \in \Gamma_i} p_{ij} x_{ij} \leq T^* \qquad (\alpha_i)$$

- Every job is assigned somewhere

$$\sum_{i \in \Gamma_j} x_{ij} = 1 \qquad (\beta_j)$$

- Non-negativity

$$x_{ij} \geq 0 \qquad (\gamma_{ij})$$

Relaxation

Many feasible solutions, add strictly convex objective $F(x)$

$$
\begin{array}{lll}
\min & F(x) & \\
\text{s.t.} & \sum_{j \in \Gamma_i} p_{ij} x_{ij} \leq T^* & \forall \text{ machines} i \\
 & \sum_{i \in \Gamma_j} x_{ij} = 1 & \forall \text{ jobs } j \\
 & x_{ij} \geq 0 & \forall i, j
\end{array}
$$

Lagrangian:

$$
F(x) + \sum_i \alpha_i \left(\sum_{j \in \Gamma_i} p_{ij} x_{ij} - T^* \right) + \sum_j \beta_j \left(1 - \sum_{i \in \Gamma_j} x_{ij} \right) - \sum_{i,j} \gamma_{ij} x_{ij}
$$

Reconstruction

Assume that $\frac{\partial F(x)}{\partial x_{ij}} = f_{ij}(x_{ij})$, for some f_{ij}
KKT yields:

$$f_{ij}(x_{ij}) + \alpha_i p_{ij} - \beta_j - \gamma_{ij} = 0$$

$$x_{ij} > 0 \implies \gamma_{ij} = 0$$

If $x_{ij} > 0$, then

$$x_{ij} = f_{ij}^{-1}(\beta_j - p_{ij}\alpha_i)$$

Thus,

$$x_{ij} = \max\{0, f_{ij}^{-1}(\beta_j - p_{ij}\alpha_i)\}$$

Conclusion: can reconstruct solution from dual variables!

Potential Algorithm

Note: Given α, can also reconstruct β online

The previous ideas lead to the following online algorithm:

1. Predict α
2. Given job j, and its values p_{ij}
    - Compute β_j from α
    - Reconstruct x_{ij}
3. Round the fractional solution online