Introduction to Algorithms with Predictions

Christian Coester (University of Oxford)

Scheduling with Restricted Assignment

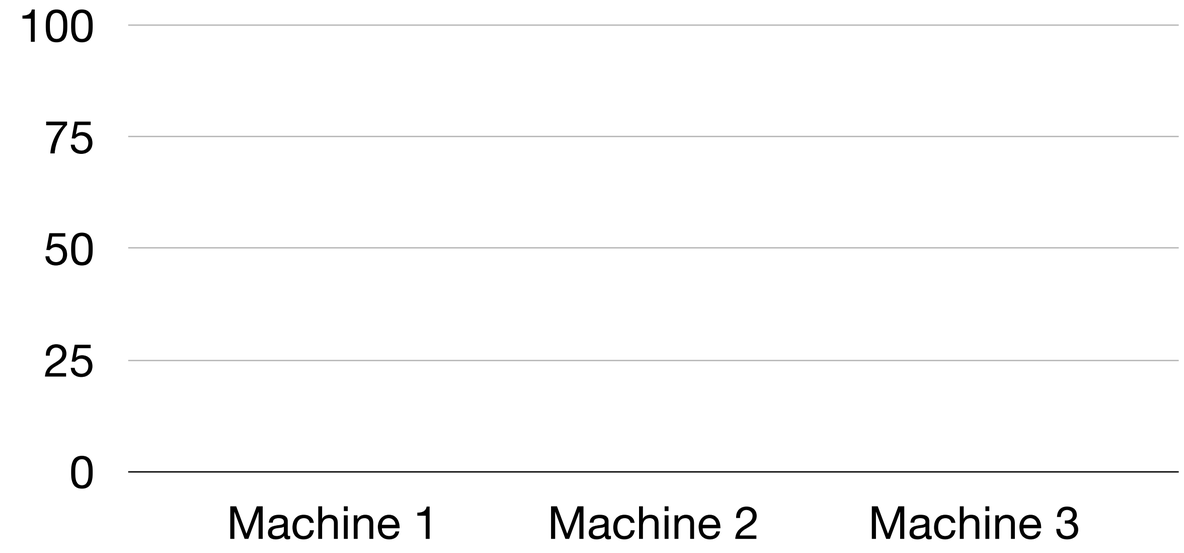
m machines

jobs $j = 1, \ldots, n$ arrive online

p_j: size of job

N_j: feasible machines

Must assign immediately



Goal: Minimize maximum load

Scheduling with Restricted Assignment

c-competitive if

$$\text{max load} \leq c \cdot \text{OPT}$$

Known: Greedy is $O(\log m)$-competitive [Azar,Naor,Rom 95]

$\Omega(\log m)$ lower bound for any online algorithm

In practice: Solve similar instances repeatedly

Can we do better with predictions?

What to Predict?

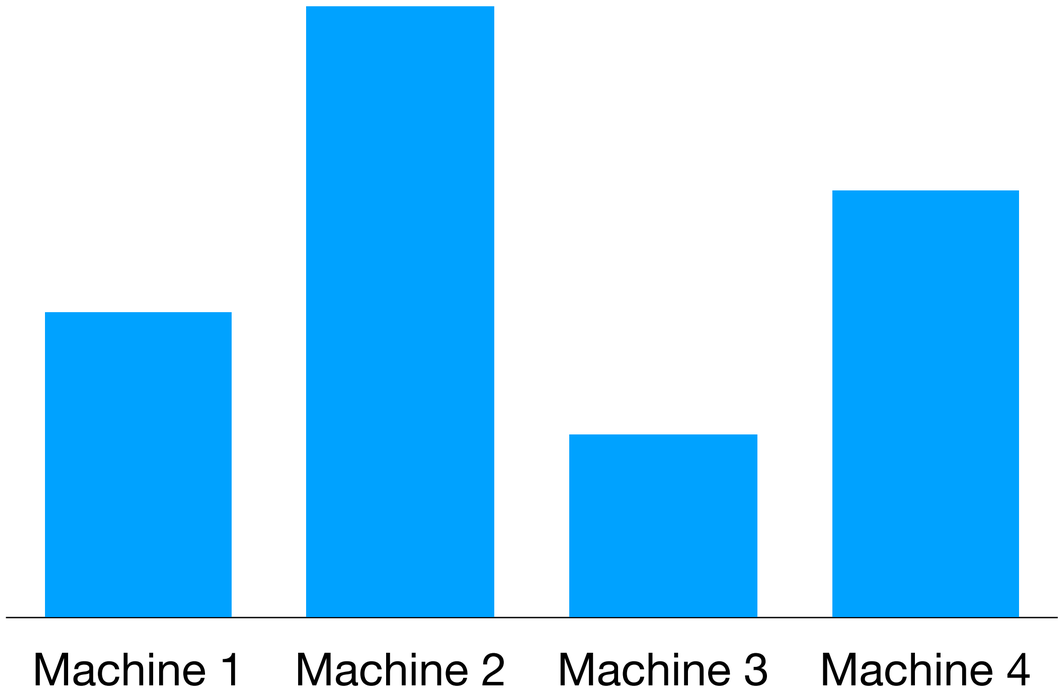
Total size for each job type

2^m job types

a lot of information, perhaps unrealistic

What to Predict?

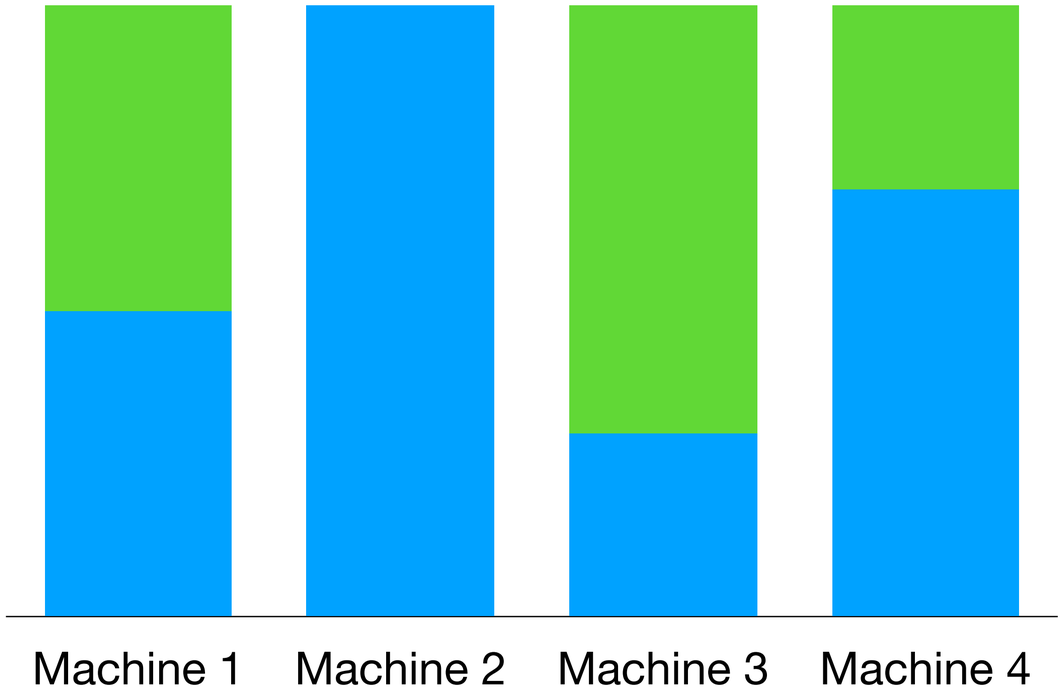
Load of machines in optimal solution



What to Predict?

Load of machines in optimal solution

dummy jobs ⇝ same loads



Goal: different measure of contentiousness of machines

What to Predict?

Predict a single weight $w_i > 0$ for each machine i

Intuition: small weight = contentious machine

Consider fractional algorithms:

$$x_{ij} = \text{fraction of job } j \text{ on machine } i$$

⇝ can be converted into randomized algorithm

[Lattanzi,Lavastida,Moseley,Vassilvitskii 20]

Existence of weights

Theorem: $\forall$ instance $\forall \epsilon > 0$ $\exists w_1, \ldots, w_m > 0$ s.t.

$$x_{ij}(w) = \frac{w_i}{\sum_{k \in N_j} w_k}$$

is $(1 + \epsilon)$-competitive.

[Agrawal,Zadimoghaddam,Mirrokni 18]

Existence of weights

Theorem: ∀ instance $\forall \epsilon > 0$ $\exists w_1, \ldots, w_m > 0$ s.t.

$$x_{ij}(w) = \frac{w_i}{\sum_{k \in N_j} w_k}$$

is $(1+\epsilon)$-competitive.

[Agrawal,Zadimoghaddam,Mirrokni 18]

Model

$$x_{ij}(w) = \frac{w_i}{\sum_{k \in N_j} w_k}$$

Receive predictions $\hat{w}_1, \ldots, \hat{w}_m$ of good weights $w_1, \ldots, w_m$

WLOG $\hat{w}_i \geq w_i$

Prediction error: $\eta = \max_i \eta_i$ where $\eta_i = \frac{\hat{w}_i}{w_i} \geq 1$

[Lattanzi,Lavastida,Moseley,Vassilvitskii 20]

Naive algorithm

$$x_{ij}(w) = \frac{w_i}{\sum_{k \in N_j} w_k}$$

$$\eta_i = \frac{\hat{w}_i}{w_i} \geq 1$$

$$x_{ij} = x_{ij}(\hat{w}) = \frac{\hat{w}_i}{\sum_{k \in N_j} \hat{w}_k}$$

This is $O(\eta)$-competitive:

$$x_{ij}(\hat{w}) = \frac{\hat{w}_i}{\sum_{k \in N_j} \hat{w}_k} \leq \frac{\eta_i \cdot w_i}{\sum_{k \in N_j} w_k} \leq \eta \cdot x_{ij}(w)$$

Can we do better?

[Lattanzi,Lavastida,Moseley,Vassilvitskii 20]

Assumption: OPT known

$$x_{ij}(w) = \frac{w_i}{\sum_{k \in N_j} w_k}$$

Initialize $L_i \leftarrow 0$ for each machine i

$$\eta_i = \frac{\hat{w}_i}{w_i} \geq 1$$

For each job j

For each $i \in N_j$

$$x_{ij} = x_{ij}(\hat{w}), \qquad L_i \leftarrow L_i + x_{ij}$$

For each i with $L_i > 2(1+\epsilon)$ OPT

$$\hat{w}_i \leftarrow \hat{w}_i/2, \qquad L_i \leftarrow 0$$

Theorem: This is $O(\log \eta)$-competitive.

[Lattanzi,Lavastida,Moseley,Vassilvitskii 20]

Claim: At all times, $\hat{w}_i \geq w_i/2$

$$x_{ij}(w) = \frac{w_i}{\sum_{k \in N_j} w_k}$$

Proof by contradiction:

Let $\hat{w}_i$ be first counterexample: $\hat{w}_i < w_i/2$

$$\eta_i = \frac{\hat{w}_i}{w_i} \geq 1$$

Just before, had $\hat{w}_i < w_i$ and $L_i > 2(1+\epsilon)$ OPT

$$x_{ij}(\hat{w}) = \frac{\hat{w}_i}{\sum_{k \in N_j} \hat{w}_k} < \frac{w_i}{\sum_{k \in N_j} w_k/2} = 2x_{ij}(w)$$

Contradiction

[Lattanzi,Lavastida,Moseley,Vassilvitskii 20]

Claim: At all times, $\hat{w}_i \geq w_i/2$

Initially: $\hat{w}_i = \eta_i w_i \leq \eta w_i$

$\rightsquigarrow$ $O(\log \eta)$ rounds per machine

Each round contributes $O(\text{OPT})$

$\Longrightarrow$ $O(\log \eta)$-competitive

$$x_{ij}(w) = \frac{w_i}{\sum_{k \in N_j} w_k}$$

$$\eta_i = \frac{\hat{w}_i}{w_i} \geq 1$$

[Lattanzi,Lavastida,Moseley,Vassilvitskii 20]

Why do good weights exist?

$$\min L$$

$$\text{s.t.} \quad \sum_j p_j x_{ij} \leq L \qquad \forall i$$

$$\sum_{i \in N_j} x_{ij} = 1 \qquad \forall j$$

$$x_{ij} \geq 0 \qquad \forall i, j$$

$$\min$$

$$\text{s.t.} \quad \sum_j p_j x_{ij} \leq (1+\epsilon)\,\mathsf{OPT} \qquad \forall i$$

$$\sum_{i \in N_j} x_{ij} = 1 \qquad \forall j$$

$$x_{ij} \geq 0 \qquad \forall i, j$$

$$\min \sum_{i,j} p_j \quad x_{ij} \ln x_{ij}$$

$$\text{s.t.} \quad \sum_{j} p_j x_{ij} \leq (1+\epsilon)\,\mathsf{OPT} \qquad \forall i$$

$$\sum_{i \in N_j} x_{ij} = 1 \qquad \forall j$$

$$x_{ij} \geq 0 \qquad \forall i,j$$

$$\min \sum_{i,j} p_j \left(x_{ij} \ln x_{ij} - x_{ij} \right)$$

$$\text{s.t.} \quad \sum_{j} p_j x_{ij} \leq (1+\epsilon)\,\mathsf{OPT} \qquad \forall i$$

$$\sum_{i \in N_j} x_{ij} = 1 \qquad \forall j$$

$$x_{ij} \geq 0 \qquad \forall i,j$$

$$\min \sum_{i,j} p_j \left(x_{ij} \ln x_{ij} - x_{ij} \right)$$

$$\text{s.t.} \quad \sum_j p_j x_{ij} \leq (1+\epsilon)\,\text{OPT} \qquad \forall i \qquad (y_i)$$

$$\sum_{i \in N_j} x_{ij} = 1 \qquad \forall j \qquad (z_j)$$

$$x_{ij} \geq 0 \qquad \forall i,j \qquad \longleftarrow \text{ never tight}$$

KKT $\implies$ $\exists y_i, z_j$ such that $p_j y_i + z_j = p_j \ln x_{ij}$

KKT $\implies \exists y_i, z_j$ such that $p_j y_i + z_j = p_j \ln x_{ij}$

$$\implies x_{ij} = e^{y_i + z_j / p_j}$$

$$\implies x_{ij} = \frac{e^{y_i + z_j / p_j}}{\sum_{k \in N_j} e^{y_k + z_j / p_j}}$$

KKT $\implies \exists y_i, z_j$ such that $p_j y_i + z_j = p_j \ln x_{ij}$

$$\implies x_{ij} = e^{y_i + z_j/p_j}$$

$$\implies x_{ij} = \frac{e^{y_i + z_j/p_j}}{\sum_{k \in N_j} e^{y_k + z_j/p_j}} = \frac{e^{y_i}}{\sum_{k \in N_j} e^{y_k}}$$

KKT $\Longrightarrow$ $\exists y_i, z_j$ such that $p_j y_i + z_j = p_j \ln x_{ij}$

$$\Longrightarrow \quad x_{ij} = e^{y_i + z_j/p_j}$$

$$\Longrightarrow \quad x_{ij} = \frac{e^{y_i + z_j/p_j}}{\sum_{k \in N_j} e^{y_k + z_j/p_j}} = \frac{e^{y_i}}{\sum_{k \in N_j} e^{y_k}} = \frac{w_i}{\sum_{k \in N_j} w_k} \quad \text{for } w_i = e^{y_i}$$

Moreover: If input comes from distribution, best predicted weights are efficiently learnable.

What if predictions are bad?

Making algorithm robust

If a machine would have load $> \log m \cdot \text{OPT}$,

switch to $O(\log m)$-competitive online algorithm

Overall: Competitive ratio $O\left(\min\{\log \eta, \log m\}\right)$

[Lattanzi,Lavastida,Moseley,Vassilvitskii 20]

Making algorithms robust — more generally

Two competing objectives

1. Let good predictions guide you
2. Don’t let bad predictions mislead you

Can we achieve both?

Often YES!

MTS (Metrical Task Systems) [Borodin,Linial,Saks 87]

Metric space (M, d)

Algorithms starts at $p_0 \in M$

At time $t = 1,2,\ldots$

$c_t : M \to \mathbb{R}_{\geq 0}$ revealed

Algorithm chooses $p_t \in M$

Pay $c_t(p_t) + d(p_{t-1}, p_t)$

Many special cases: k-server, caching, layered graph traversal, convex function chasing, dynamic power management, ski rental, …

MTS (Metrical Task Systems)

Metric space (M, d)

Algorithms starts at $p_0 \in M$

At time $t = 1,2,\ldots$

$c_t : M \to \mathbb{R}_{\geq 0}$ revealed

Algorithm chooses $p_t \in M$

Pay $c_t(p_t) + d(p_{t-1}, p_t)$

$M = \{\text{rent}, \text{buy}\}$

$d(\text{rent}, \text{buy}) = \text{buying cost}$

$p_0 = \text{rent}$

$c_t(\text{rent}) = 1$

$c_t(\text{buy}) = 0$

Many special cases: k-server, caching, layered graph traversal, convex function chasing, dynamic power management, ski rental, …

Theorem: Algorithms A and B for any MTS can be combined online into an algorithm C s.t.

$$\text{cost}_C \leq 9 \cdot \min\{\text{cost}_A, \text{cost}_B\}.$$

Application:

A non-robust learning-augmented algorithm

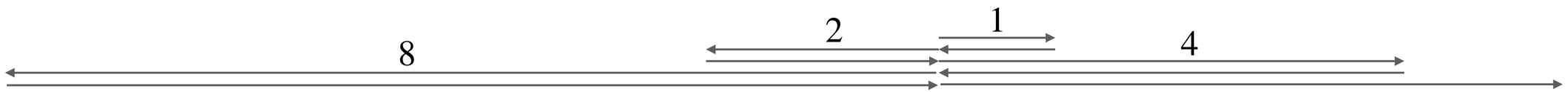
B online algorithm (ignores predictions)

$\Longrightarrow$ C robust learning-augmented algorithm

[Fiat,Rabani,Ravid 94]

[Antoniadis,Coester,Elias,Polak,Simon 20]

Cow-path problem



cost $\leq 2(1 + 2 + \ldots + 2^n) + \text{OPT}$ where $2^{n-1} < \text{OPT} \leq 2^{n+1}$

Cow-path problem

$$\text{cost} \le 2(\underbrace{1 + 2 + \ldots + 2^n}_{< 2^{n+1} < 4 \cdot \text{OPT}}) + \text{OPT} \qquad \text{where } 2^{n-1} < \text{OPT} \le 2^{n+1}$$

Cow-path problem

$\text{cost} \leq 2(\underbrace{1 + 2 + \ldots + 2^n}_{< 2^{n+1} < 4 \cdot \text{OPT}}) + \text{OPT}$ where $2^{n-1} < \text{OPT} \leq 2^{n+1}$

$\leq 9 \cdot \text{OPT}$

Theorem: Algorithms A and B for any MTS can be combined online into an algorithm C s.t.

$$\text{cost}_C \leq 9 \cdot \min\{\text{cost}_A, \text{cost}_B\}.$$

Algorithm C:

for $n = 0,1,2,\ldots$

$$F = \begin{cases} A, & n \text{ even} \\ B, & n \text{ odd} \end{cases}$$

while $\text{cost}_F \leq 2^n$

follow F

Analysis:

In iteration n:

pay $\leq 2^n$ to follow F

pay $\leq 2^n$ to return to p_0

$2^{n-1} < \min\{\text{cost}_A, \text{cost}_B\} \leq 2^{n+1}$

[Fiat,Rabani,Ravid 94]

[Antoniadis,Coester,Elias,Polak,Simon 20]

A better randomized combination

Theorem: Given $\epsilon > 0$ and algorithms $A_1, \ldots, A_m$ for an MTS of diameter D,

$\exists$ algorithm with expected cost

$$\leq (1+\epsilon) \cdot \min_i \text{cost}_{A_i} \quad + \quad O\left(\frac{D \log m}{\epsilon}\right)$$

[Blum,Burch 00]

A better randomized combination

Algorithm maintains probability distribution $p = (p_1, \ldots, p_m)$ over algorithms

$\rightsquigarrow$ Follow A_i w.p. p_i

where

$$p_i = \frac{w_i}{\sum_j w_j} \quad \text{and} \quad w_i = \left(1 - \frac{\epsilon}{2}\right)^{\text{cost}_{A_i}/D}$$

When p changes to p', switching costs $\leq D \cdot \sum_{i=1}^{m} \max\{p'_i - p_i, 0\}$

[Blum,Burch 00]

Corollary: For any problem that can be modeled as an MTS, if there is

a c-competitive algorithm without predictions

an $f(\eta)$-competitive algorithm when given predictions with error η

then there is an algorithm with competitive ratio

$9 \cdot \min\{c, f(\eta)\}$ deterministically

$(1+\epsilon) \cdot \min\{c, f(\eta)\}$ randomized up to additive cost $O(D/\epsilon)$

[Antoniadis,Coester,Elias,Polak,Simon 20]