Last time:

We talked about potential (general congestion) games, where solutions that minimizes Φ are Nash equilibriums. If we are lucky, Φ can be minimized in poly-time and a Nash will be found. Commonly, minimizing Φ is NP-complete, which we cannot do. However, we don't need the particular Nash solution minimizing Φ, we just need one.

Reviewing, a Nash is the same as the local optimum of Φ.

The meaning of local optimum is ill-defined, but in this context we can take it to mean that no single player can decrease the value of Φ

Theorem: (Fabrikant, Papadimitriou; Talwar)

Finding a Nash in general potential (and routing) games is PLS-complete. PLS complete means that the an local solution to a problem can be found in polynomial time and that it belongs to a fundamental class of such problems.[1]

Although these can be difficult, we don't need to give up. Instead, we can try novel approaches. Like NP-completeness, this is an open-problem so there are other things we can attempt.

Ideas:

1st,Use players natural steps to decrease Φ

Here we let one-player-at-a-time, player *j,* switch to their best strategy (decreasing Φ).

Pros/Cons

+ if player *j* changes → Φ decreases

+ if no-one changes, then we are at local minimum

- decreases can be very slow (bad convergence)

Maybe we can do better speed-wise

Optimized version to increase speed:

We choose a tolance $\varepsilon > 0$, which we use to limit players moves to only those that are greater than ε. In otherwords, there is a minimum improvement a player must make to change.

[1] See D.S. Johnson, C.H. Papadimitriou, and M. Yannakakis. ***How Easy Is Local Search?*** Journal of Computer and System Sciences, 37:79-100, 1988.

So players only take a step if Φ decreases by a (1- ε) factor

Pros/Cons

+ if player *j* changes → Φ decreases

- no longer produces local-optimum

+ faster

So we have a faster algorithm, but with some bound on how good it is.

Claim: The number of such improving steps possible is bounded by: $\varepsilon^{-1} \log \frac{\Phi_{\max}}{\Phi_{\min}}$.

Proof: $(1-\varepsilon)^K$ is the factor decrease in Φ after *K* steps where ($K = \varepsilon^{-1}$). Realize that $(1-\varepsilon)^{\varepsilon^{-1}} \sim 1/e$

N.B. For this claim and proof, assume $\Phi \geq 0$, which means that this claim only has meaning for certain Φ functions.

Definition: Approximate Nash solutions with tolerance ε is an ε-Nash if no player *j* can change strategy and save ε factor of cost (e.g. decrease $s_j - t_j$ path delay be (1- ε) factor)

Question: Does a big-step local search generate ε-Nash?

$\exists\ \varepsilon$ s.t. ε big steps → δ Nash Big $\varepsilon = \delta$ n n=# players

There is no definition where this can happen

2nd, Alternate Idea

Lets not try finding a Nash. A Nash is beneficial in avoiding "The price of anachy", however they can be hard to find. But maybe we don't need a Nash? We can give players a few chances to move, and hopefully afterwards, the state will be an good approximation for a Nash.

Variants:

Players can move using schedules like: Round robin, only one moves all the time, and give everyone one chance to move.

For figure 1, Job *j* has load p_j and can go on any machine

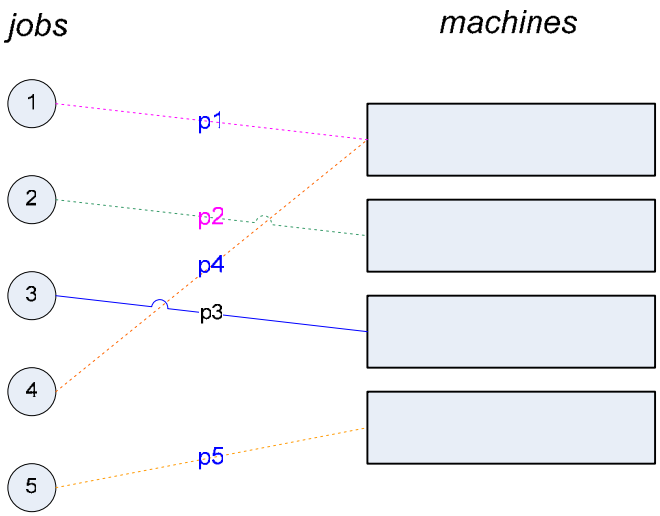

Figure 1 - load balancing jobs 1 … *n* on *m* uniform machines

Objective: We want to lower the make span or minimize the max. load. Figure 2 shows a bad Nash

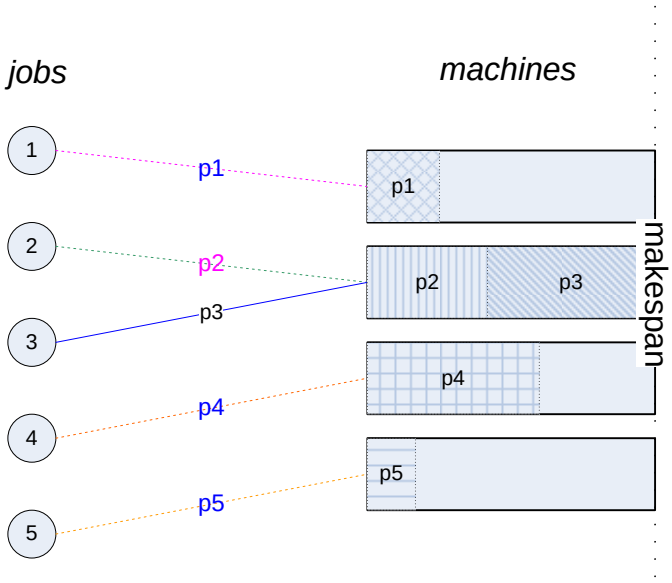

Figure 2

Theorem:

The maxload for a Nash is within a factor of 2 for the minimum possible maxload. maxload Nash ≤ 2 * (min possible max load)

Next Time:

(Thm.) if all *j* had a chance to move → max load ≤ 2 minimum maximum load.