1 Existence of Nash Equilibria for Load Balancing

We consider the load balancing problem, which has the following input:

- m machines, a continuous and monotone increasing function $r_i(L)$ giving the response time of machine i as a function of its load L,
- n job types, where p_j is the total load of type j, and $S_j \subset \{1, \ldots, m\}$ is the set of machines on which type j can be scheduled.

A solution to the load balancing problem is an assignment x satisfying

$$(SOL) \qquad \begin{aligned} & x_{ij} \geq 0, \text{ for all } i, j \\ & \sum_{i=1}^{m} x_{ij} = p_j \text{ for all } j \\ & x_{ij} = 0 \text{ if } i \notin S_j \\ & \text{load } L_i = \sum_{j=1}^{n} x_{ij} \text{ for all } i \end{aligned}$$

We defined a Nash equilibrium as a choice of action by each player so that no player can improve his/her value by changing his/her action alone. For the load balancing problem, we showed that x is a Nash equilibrium if

$$\text{for all } x_{ij} > 0 \text{ and } k \in S_j \Rightarrow r_i(L_i) \leq r_k(L_k)$$

Last time we showed that we can find a Nash equilibrium via a sequence of maximum flow computations. Today we wil show that we can find a Nash equilibrium via a single optimization, and generalize this to networks.

1.1 Discrete & uniform jobs

We will start by considering **discrete and uniform jobs**, i.e. p_j is integer, and we have p_j unit size jobs. A solution in this case is given by x satisfying the conditions in (SOL) and

$$(*) \qquad x_{ij} \text{ integer for all } i, j$$

Let

$$\Phi = \sum_{i=1}^{m} \sum_{\xi=1}^{L_i} r_i(\xi)$$

We call Φ a **potential function** and we will show that when a job changes from one machine to another, Φ tracks the change in response time *for that job type*.
In general we call games such that there exists a potential function that tracks a player's change in utility, **potential games**.

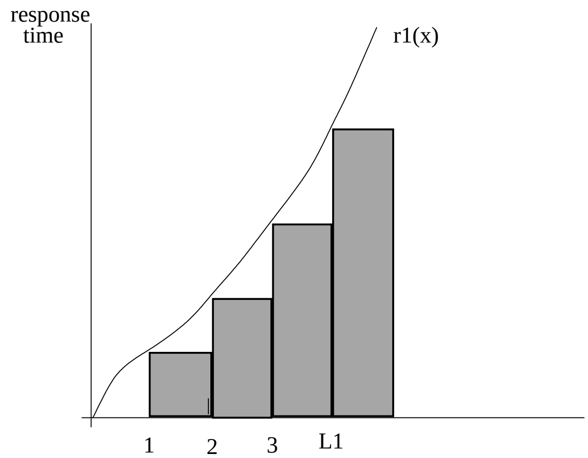

Figure 1: The shaded area is $\sum_{\xi=1}^{L_1} r_1(\xi)$.

Theorem 1 *If one (unit-size) job of type j changes from machine i to machine k, the decrease in response time for job j is equal to the decrease in Φ.*

Proof. Since one unit-size job is moved from machine i to machine k, job j's total response time decreases by $r_i(L_i) - r_k(L_k+1)$. Clearly, the decrease in Φ is also exactly equal to $r_i(L_i) - r_k(L_k+1)$. ■

Corollary 2 *The existence of the potential function Φ implies*

(i) Starting from any state, we can find a Nash equilibrium in finite (possibly exponential) time.

(ii) A solution with minimum Φ-value is a Nash equilibrium.

Proof. If we allow jobs to switch one-at-a-time if it improves their response time, Φ will decrease until no job can improve its response time by switching, i.e. until we find a Nash equilibrium. Since there are only a finite number of ways of assigning the jobs to the machines (since we are assuming discrete jobs), and we cannot cycle because Φ decreases in each iteration, this gives a finite time algorithm for finding a Nash equilibrium.

■

Note that not all Nash equilibria minimize Φ. See an example from the lecture of August 29 in Figure 2.

1.2 Generalization to continuous case

We now drop the integrality requirement $(*)$ on the solution and consider p_j as consisting of inifinitesimally small jobs.

Based on the previous section, a natural candidate for the potential function is

$$\Phi = \sum_{i=1}^{m} \int_0^{L_i} r_i(\xi) d\xi$$

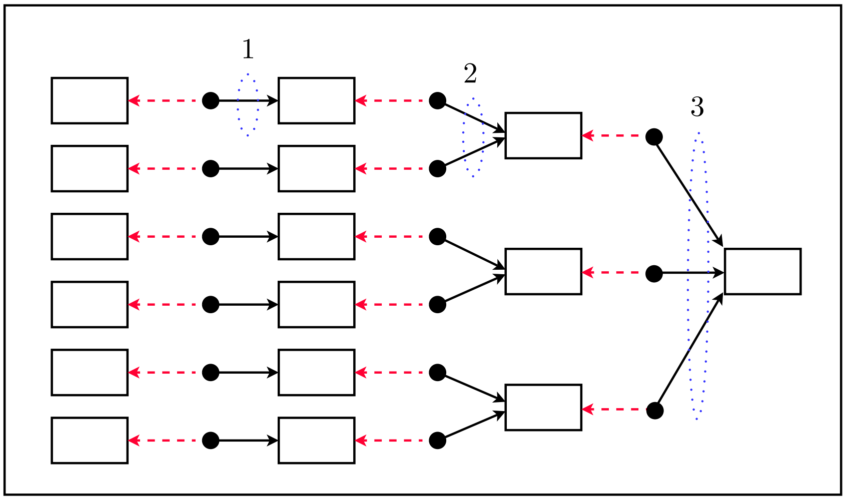

Figure 2: If $r_i(x) = x$ for all machines i, both the red dashed assignment and the black solid assignment are Nash equilibria, but Φ("red dashed ") = 15, Φ("black solid") = 21.

The following theorem states that Φ has the desired quality, i.e. it tracks the change in response time for job j if j shifts a small amount to another machine.

Theorem 3 *If job j has $x_{ij} > 0$ and $k \in S_j$ and the Nash conditions are not satisfied, i.e. $r_i(L_i) > r_k(L_k)$ then Φ decreases when we shift a small amount from x_{ij} to x_{kj}.*

Proof. We know that if

$$\frac{\partial \Phi}{\partial x_{ij}} > \frac{\partial \Phi}{\partial x_{kj}}$$

then there exists some $\epsilon > 0$ such that removing ϵ from x_{ij} and adding it to x_{jk} decreases Φ.

Now

$$\frac{\partial \Phi}{\partial x_{kj}} = \frac{\partial \Big(\int_0^{L_k} r_k(\xi) d\xi \Big)}{\partial x_{kj}} = r_k(L_k)$$

(where we use continuity of r_k in the last equality), and similarly $\frac{\partial \Phi}{\partial x_{ij}} = r_i(L_i)$, hence the fact that the Nash conditions are not satisfied implies that $\frac{\partial \Phi}{\partial x_{ij}} > \frac{\partial \Phi}{\partial x_{kj}}$. ■

Corollary 4 *The existence of the potential function Φ implies that*

(i) A solution with minimum Φ-value is a Nash equilibrium.

(ii) A Nash equilibrium exists.

(iii) We can find a Nash equilibrium in polynomial time.

Proof. It follows immediately from Theorem 3 that a solution with minimum Φ-value (if it exists) must be a Nash equilibrium. Since Φ is a continuous function (as we are assuming the $r_i(x)$ are continuous for all i), and since the set of all feasible solutions is closed and bounded, Φ does indeed achieve a minimum on the set of feasible solutions, so a Nash equilibrium does exist.

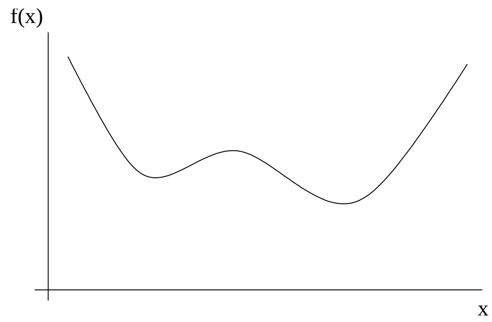

Figure 3: The function $f(x)$ is not convex.

To show that we can find a Nash equilibrium in polynomial time, we need the following facts about convexity and convex programming:

1. A differentiable function of one variable is (strictly) convex on an interval if and only if its derivative is (strictly) monotone increasing on that interval.

   If two functions f and g are (strictly) convex, then so is $f + g$.

2. We can minimize a convex function over a convex set in polynomial time.

For any i, $\frac{\partial \int_0^{L_i} r_i(\xi)d\xi}{\partial L_i} = r_i(L_i)$ and $r_i(L_i)$ is montonone increasing, so by the first fact the function $\int_0^{L_i} r_i(\xi)d\xi$ is convex and hence $\Phi = \sum_{i=1}^{m} \int_0^{L_i} r_i(\xi)d\xi$ is also convex.

The set of feasible loads $L = (L_1, \ldots L_m)$ is convex, since for any feasible loads (i.e. loads such that there exists a feasible assignment x of the jobs that results in those loads) L^1 and L^2, the load $\lambda L^1 + (1 - \lambda)L^2$ is also feasible for any $0 \leq \lambda \leq 1$. To see this, let x^1 be an assignment of jobs that gives rise to loads L^1, and let x^2 be an assignment of jobs that gives loads L^2. It is straightforward to check that $\lambda x^1 + (1-\lambda)x^2$ obeys the constraints in (SOL) and gives loads $\lambda L^1 + (1-\lambda)L^2$. Hence by the second fact, we can find a Nash equilibrium (a solution of minimum Φ-value) in polynomial time. ■

Note that any local minimum of a convex function is also a global minimum (see Figure 3). A strictly convex function has a unique minimum. Hence any Nash equilibrium minimizes Φ, and if $r_i(L)$ is strictly monotone for all i, then Φ has a unique minimum and there is a unique Nash equilibrium. Combining this observation with Theorem 3, we get the following theorem.

Theorem 5 *x is a Nash equilibrium if and only if x minimizes $\Phi = \sum_{i=1}^{m} \int_0^{L_i} r_i(\xi)d\xi$.*