CS294-1 Algorithmic Aspects of Game Theory **Spring 2001**

Final project: The Price of Anarchy

Dror Weitz, 14580254

1 Introduction

A key problem in network management is that of routing traffic in order to optimize network performance. Given the traffic to be routed, we wish to find a routing that optimize certain performance parameters, e.g., the average travel time. Solutions to these kind of problems can be extremely hard since the latency experienced by each edge usually depends on the amount of traffic traveling through that edge. In practice, a common solution is to let the individual agents behave selfishly, i.e., choose the minimum latency path available to them, hoping that such behavior will also optimize the total network performance. Recently, a few papers addressed the question of *how far can a Nash equilibrium, i.e., obtained by selfish behavior, be from the optimal routing?*

Koutsoupias and Papadimitriou [KP99] introduced this question. They only considered networks with a single pair of source and sink, with parallel edges between the two, and linear latency functions for each edge. They compared the *maximum latency* experienced by an agent in the the worst Nash equilibrium to the *maximum latency* under an optimal routing. They showed that if there are only two parallel edges, then the ratio between the *maximum latency* in a Nash equilibrium and in an optimal routing is at most $\frac{3}{2}$ if the edges have the same latency functions, and that this bound is tight. They also showed a lower bound equal to the golden ratio for the case of different latency functions on each edge. For networks with more links, they give a lower bound of $\Omega(\frac{\log m}{\log \log m})$ and an upper bound of $O(\sqrt{m \log m})$ on the ratio between the *maximum latency* in a Nash equilibrium and in an optimal routing.

Later, Roughgarden and Tardos [RT00] considered a general network, where traffic may be required to be routed from any node to any node, and with arbitrary non-decreasing continuous latency functions. However, they required that there are infinite number of agents, i.e., each agent controls a negligible part of the flow, or, alternatively, that each agent can split his flow between different paths with arbitrary precision. The network parameter (or social welfare function) that is considered in [RT00] is the *total latency* experienced by the system, or equivalently, the *average latency* an agent is experiencing. They showed that when the latency functions of the edges are linear, then the *average latency* in a Nash equilibrium is at most $\frac{4}{3}$ times the *average latency* in an optimal routing. For arbitrary latency functions, they give a bicriteria, where the *average latency* in a Nash equilibrium is at most twice the *average latency* in an optimal routing of double the amount of flow.

After defining the exact model in section 2, I will review the results and key ideas in the proofs of [RT00] in section 3. In section 4, while still in the model framework of [RT00], I suggest looking at other network parameters to optimize. I give an example where the *maximum latency* in a Nash equilibrium can be arbitrary large (depending on the size of the network) in comparison to the *maximum latency* in an optimal routing, even when the latency functions are linear. This example is different in nature from the lower bound in [KP99] since the latter comes from the unsplittable nature of the problem in their setting, where the example I give is for the setting in [RT00] where each agent controls a negligible part of the flow. I also mention speed (rather than latency) parameters, which may be worth researching as future work.

2 The Model

I now describe the model and notations as defined in [RT00]. Consider a directed network $G = (V, E)$ with k source-destination vertex pairs $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$. With each such pair we associate a positive *rate* r_i, the amount of flow to be routed from s_i to t_i. Each edge $e \in E$ has a load dependent latency that is denoted by $l_e(\cdot)$. We assume that l_e is non-negative, continuous and nondecreasing. A legal flow f for this network is an assignment to simple paths s.t. for every i, the sum of flow along all paths from s_i to t_i is r_i, and no flow is assigned to paths between other vertices. We denote by f_e the sum of flow along the edge e. As mentioned before, $l_e(f_e)$ is the latency experienced along the edge e in the flow f. The latency of a path is just the sum of latencies of the edges in the path.

We assume that the rate to be routed between each pair $s_i - t_i$ is the result of aggregating over infinitely many agents, each contributing a negligible amount to this rate. Each such agent wishes to minimize its travel time, and therefore, given the congestion resulting from the routing of the rest of the agents in the system, the agent will choose the minimum latency path available to him. Since each agent controls a negligible amount of the flow, w.l.o.g. we can restrict the agents to choose only *pure strategies*. A Nash equilibrium is then a routing of all the agents in the system where each agent travels along the minimum latency path available to him given the routing of the other agents. Equivalently, it is a flow f where, given the congestion of f, for every i, any $s_i - t_i$ path that carries a positive amount of flow is minimum latency path (among $s_i - t_i$ paths). A direct consequence is that if f is a flow at Nash equilibrium, then for every i, all $s_i - t_i$ paths that carry a positive amount of flow have the same latency. Beckman et al. [BMW56] observed that a Nash equilibrium is an optimal solution to a related convex problem, thus showing the existence and essential uniqueness of such equilibria.

3 Optimizing the Average Latency

The network parameter that is compared in [RT00] is the *total latency* experienced by the system, or equivalently, the *average latency* experienced by an agent in the system. Let $C(f)$ denote this total latency, which we also call the cost of the flow f. $C(f)$ equals to the summation over paths of the latency along a path times the flow routed through it. It is easy to see that in fact $C(f) = \sum_{e \in E} l_e(f_e) f_e$. The analysis in [RT00] bounds the ratio $\rho = \frac{C(f)}{C(f^*)}$ where f is a Nash flow and f^* is an optimal flow with respect to C.

3.1 General Latency Functions

When assuming no restriction on the latency functions of the edges (expect that they are non-negative, nondecreasing, and continuous) there are examples where this ratio is arbitrarily large even for a network with only two edges. However, Theorem 3.1 of [RT00] states an interesting *bicriteria* where the Nash flow is compared to the optimal flow of the same network with larger rates. They show that if f is a Nash flow for the given network, then for any flow f' for the same network, but with rates $(1+\delta)r_i$, $C(f) \leq \frac{1}{\delta} C(f')$. In particular, $C(f) \leq C(f')$ if the rates are doubled in f', i.e., the average latency in f is at most twice that of f'.

The proof makes an important use of the fact that if a flow f is in Nash equilibrium then all $s_i - t_i$ paths that carry a positive amount of flow are minimum latency paths (among $s_i - t_i$ paths) in f. Another property that is used is the fact the latency functions are non-decreasing, and therefore routing $(1+\delta)$ times flow along the same path must increase the total latency of this path (the latency times the flow going through this path) by a factor of at least $(1+\delta)$. I give some more details of the proof below.

First, they construct a new set of edge latency functions $\overline{l}_e$. The new latency functions make sure that in any flow the latency of any edge is at least as much as it is in the Nash flow f under the original latency functions l_e. They do this by introducing a cut-off point to the original functions s.t. $\overline{l}_e = l_e$ for values above f_e, but equals to $l_e(f_e)$ for values below that. It is now easy to see that for any flow f', even if it routes different total rates than f does, $\overline{C}(f') - C(f') \leq C(f)$ where $\overline{C}$ is the cost w.r.t. to $\overline{l}$. This is because differences in the latency can only occur in edges that carry less than f_e flow, and since in these cases the difference in the latency can be at most $l_e(f_e)$.

As mentioned before, the latency of any path in f' under $\overline{l}$ is at least as much as the latency of the same path in f under l. Since f is a Nash flow for l, then each flow carrying $s_i - t_i$ path is a minimum latency path (among all $s_i - t_i$ paths). Therefore, the latency of any $s_i - t_i$ path in f' under $\overline{l}$ is at least as much as the latency of any flow carrying $s_i - t_i$ path in f under l. Now, since f' routes $1+\delta$ times more rate then f, it must be that the total latency from all $s_i - t_i$ paths in f' under $\overline{l}$ is at least $(1+\delta)$ times the total latency from all $s_i - t_i$ paths in f under l. Therefore, $\overline{C}(f') \geq (1+\delta)C(f)$. Combining with the fact that $\overline{C}(f') - C(f') \leq C(f)$ we get that $C(f) \leq \frac{1}{\delta}C(f')$.

3.2 Linear Latency Functions

In the case that all the latency functions are linear, i.e. $l_e(f_e) = a_e f_e + b_e$, the ratio ρ can be upper bounded by $\frac{4}{3}$ as stated in Theorem 4.5 of [RT00]. They also give examples of networks for which $\rho = \frac{4}{3}$. In one of the examples the network is minimal in the sense that it consists of one pair of vertices with two parallel edges between them.

The proof of Theorem 4.5 is based on looking at the *gradient* of the cost (total latency) along a path, i.e., the marginal cost of increasing the flow along the path. The marginal cost of increasing the flow along a path P is: $(\sum_{e\in P} l_e(f_e)f_e)' = \sum_{e\in P}[l_e(f_e) + l'_e(f_e)f_e]$. In particular, when the latency functions are linear, the gradient along a path is equal to $\sum_{e\in P} 2a_e f_e + b_e$. The importance of this fact is that the latency along a path in f equals to the *gradient* along that path in $\frac{f}{2}$. Thus, if f is Nash, and the latency along a flow carrying $s_i - t_i$ path is $L_i(f)$, i.e., every other $s_i - t_i$ path has at least that latency, then every $s_i - t_i$ path has a gradient of at least $L_i(f)$ in $\frac{f}{2}$ whereas paths that carry a positive amount of flow in f have exactly that gradient. Since the cost of a path (the total latency along that path) is a convex function of the flow along that path, then the cost along a path P after changing the amount of flow along P by δ is lower bounded by the original cost plus δ times the gradient at the original point (δ might be positive or negative). This implies that for any flow f' that routes the same rates as f, $C(f') \geq C(\frac{f}{2}) + \frac{1}{2}C(f)$. This is because in going from $\frac{f}{2}$ to f' we have to add a total of $\frac{r_i}{2}$ amount of flow along paths with gradient at least $L_i(f)$. Thus, adding at least $\frac{\sum_i r_i L_i(f)}{2} = \frac{1}{2}C(f)$ to the cost of the flow. Notice that there can be paths that carry less flow in f' than in $\frac{f}{2}$, but those paths have gradient exactly $L_i(f)$ because they are flow carrying paths in f, and therefore the bound is correct. Now, clearly, $C(\frac{f}{2}) \geq \frac{1}{4}C(f)$ because we route half the flow along paths with at least half the latency. Therefore, $C(f') \geq \frac{3}{4}C(f)$ as required.

3.3 Extensions

Some of the assumptions in the model so far may be unrealistic, as discussed in [RT00]. The first assumption they look into is that each agent can accurately decide which path has minimum latency. They relax this assumption by letting each agent distinguish between paths whose latency differ by at least a factor of $(1+\epsilon)$. They then adopt the definition of a Nash equilibrium to this setting by saying that a flow is at *ϵ-approximate Nash Equilibrium* iff each agent can improve its latency by a factor of at most $(1+\epsilon)$, equivalently, iff every flow carrying $s_i - t_i$ path has latency at most $(1+\epsilon)$ times that of any other $s_i - t_i$ path. They show that the bicriteria for general latency functions can be adopted to this setting with an additional factor of $(1+\epsilon)$. They do not mention a bound for linear latency functions in this setting, but as far as I see it, the proof can be easily generalized to this setting, with the ratio being $\frac{4(1+\epsilon)}{3-\epsilon}$. This can be done by taking $L_i(f)$ as the minimum latency of any $s_i - t_i$ path in f, thus it is still a lower bound on the gradient in $\frac{f}{2}$. On the other hand, any flow carrying path in f has latency at most $(1+\epsilon)L_i(f)$, and therefore, any path with negative change can have a gradient of at most $(1+\epsilon)L_i(f)$. We can complete the proof in the same manner as before by taking $\sum_i (1+\epsilon)r_i L_i(f)$ as an upper bound on $C(f)$.

The other assumption they examine is the fact that each agent controls a negligible amount of flow. They consider the case where there are finitely many agents, each controls a strictly greater than zero amount of flow. They differentiate between two cases. In the first, the agents can split their flow between different paths. This case is similar to the original model, and they show that the bicriteria holds in this case as well.

In the second case, each agent must send his flow along one path. In this case the bicriteria no longer holds, but they give a new bicriteria for this case where the bound takes into account the maximum increase in latency over an edge that can occur by adding the flow of one agent to this edge.

4 Other Social Choice Functions

So far we examined the results in [RT00] which consider the *average latency* as the parameter to optimize and compare. However, it is interesting to compare other parameters of the network. We now consider the same network model (each agent controls a negligible amount of the flow), but look at different network parameters, comparing a parameter when it is optimized to the same parameter in a Nash flow. As before, we look at the ratio ρ between the two cases.

4.1 Maximum Latency

An example of an interesting network parameter is the *maximum latency* that any agent experiences, i.e., the maximum latency over any flow carrying path. This is in fact the parameter considered in [KP99].

First, we notice that the bicriteria proof can not be applied in this case because it is no longer true that if we increase the flow along a path by some factor then the cost of that path increases by at least the same factor (we no longer multiply by the amount of flow on the path).

As for bounding the ratio ρ between flows of the same size, there are different results depending on whether there is only one pair of source and sink or more. In case there is only one source and sink, then any upper bound on ρ for the *average latency* is an upper bound on ρ for the *maximum latency*. This is because, in this case, the average and maximum latency in a Nash flow are the same, since all flow carrying paths have the same latency. On the other hand, for any flow, the maximum latency is at least the average latency. Therefore, if all the latency functions are linear, and if there is only one pair of source and sink, then $\rho \leq \frac{4}{3}$ when considering the *maximum latency*. This bound is tight since Braess's Paradox (Figure 1 in [RT00]) is an example with $\rho = \frac{4}{3}$ for both average and maximum latency. It is worth mentioning here that [KP99] give a lower bound of $\Omega(\frac{\log m}{\log \log m})$ where $m = |E|$ even when there is only one pair of source and sink. This is not a contradiction to the above since they consider a different model where each agent controls a strictly positive amount of the flow, and can not split it (but can choose a mixed strategy). Their lower bound is based on this unsplittable nature of the flow in their model by looking at the load-balancing aspect of the problem.

When there are two or more distinct source-sink pairs, the above upper bound is no longer true. The reason is that now, a Nash flow might have very different maximum and average latencies, where the latency for one pair is much larger than the latency of the other pair, and thus, an optimal flow can do much better.

Figure 1 is an example of a network with two source-sink pairs where $\rho = \Omega(|E|)$. The latency functions on the top (v_i, w_i) edges are $l_e(f_e) = nf_e$ while on the bottom (v_i, w_i) edges it is $l_e(f_e) = f_e$. All the other edges have zero latency. We consider the case where $r_1 = 1$ and $r_2 = n^3$.

We first describe the optimal flow f^* for this example. In f^*, the flow between s_1 and t_1 is routed along the top edges. The flow between s_2 and t_2 is divided evenly between n paths, one for each v_i. From v_i the flow continues through the bottom edge to w_i, and then goes to t_2. The congestion in f^* is as follows. The top edges carry all the flow from s_1 to t_1 (one unit) and nothing else. Thus, the latency of each top edge is n, and the latency of flow from s_1 to t_1 is n^2 (summing over all these edges). Each bottom edge carries $\frac{1}{n}$ of the flow from s_2 to t_2, i.e., n^2 units, and therefore the latency of each such edge is n^2 which is also the latency of any flow from s_2 to t_2. Thus, the optimal *maximum latency* is n^2.

We go on to describe a Nash flow f in this network. In f, as before, the flow between s_2 and t_2 is divided evenly between n paths, one for each (v_i, w_i). However, the flow between v_i and w_i is split in a different

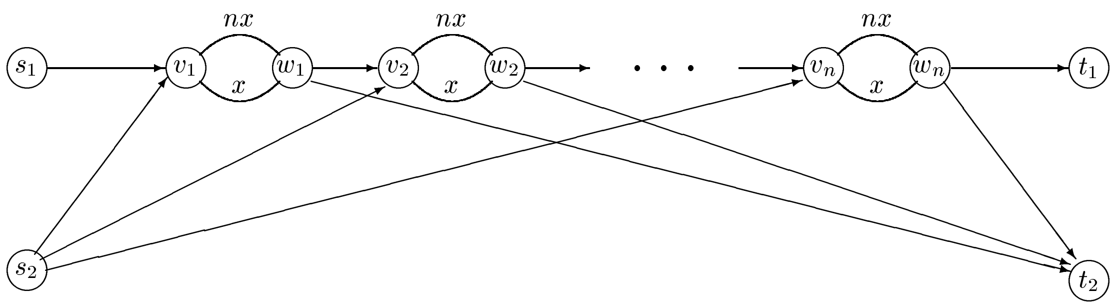

Figure 1: A Bad Example for Maximum Latency

manner. In a Nash flow, the top and bottom edges must have the same latency. Thus, the total of the flow coming from s_1 and the portion coming from s_2 are divided in such a way that the latency along the top and bottom edges is the same. More precisely, along each top edge we send $\frac{1}{n+1}(1+n^2)$ units of flow while on each bottom edge we send $\frac{n}{n+1}(1+n^2)$ units, and thus, the latency between v_i and w_i is $\frac{n}{n+1}(1+n^2)$. Since the flow from s_1 to t_1 goes through all the (v_i, w_i) pairs, the latency of this flow is $\frac{n^2}{n+1}(1+n^2) = \Omega(n^3)$. Therefore, $\rho = \Omega(n)$.

I think it might be possible to upper bound ρ in terms of $|E|$, probably even by $O(|E|)$, but I did not succeed in proving it so far. A possible path to follow is trying to show that the optimal maximum latency must be at least the latency of any edge in the Nash flow.

4.2 Average/Maximum Speed

The last example brings up another issue with the social choice functions (the network parameters) we examined so far when there are two or more source-sink pairs. The problem is that when we calculate the average or maximum latency, we treat the agents going along one source-sink pair in the same way as we treat the agents that go along another source-sink pair. It might be that the distance between one pair is much larger than the distance between the other. In this case, one might argue that these social choice functions are unfair to the agents that have to travel only a short distance. A solution might be to optimize to average or maximum *speed* of an agent, i.e., normalize the latency of each agent by the distance he has to travel before aggregating over the agents. If we wish to do that we face another question, *how should we set the distance for each source-sink pair?* The most general solution is when each edge is assigned some distance d_e and taking the distance as the shortest path distance under d_e.

If we allow any set of distances d_e, then the worst case ρ might only increase since we can always make all the pairs have the same distance without changing the congestion by adding edges with zero latency. It is also not too difficult to see that even the *average latency* result can break when normalizing by arbitrary distances, i.e., for *average speed*, ρ may be arbitrarily large. Take the example in figure 1 and assign infinite distance to the edges (s_2, v_i). The effect is that now *average speed* only considers the latency of flow between s_1 and t_1, and we can optimize that to have arbitrarily smaller latency than in Nash if r_2 is large enough.

A more interesting choice for the distances d_e is when they depend on the topology of the network and the latency functions. For example, we might set $d_e = l_e(0)$, i.e., the distance is the shortest latency path when no one is using the network. Another example is setting the distances to the latency experienced in the Nash flow, i.e., in the Nash flow, all agents (from all source-sink pairs) have speed 1. In this case, the meaning of ρ is the best possible (over flows) average/least speed (over all agents) w.r.t. to the Nash flow. From all the network parameters discussed so far, the last seems as the most interesting to me. It is worthwhile, I believe, to try getting an upper bound on ρ for this parameter. Notice the difference between this last parameter and the original non-normalized parameter. The latter considers the speed-up of the average/maximum latency while the former considers the average/maximum speed-up.

5 Summary

In this report I reviewed the results in [RT00] which gave an exact bound on the ratio between the *average latency* experienced by an agent in a Nash flow and in an optimal flow when all the latency functions are linear. In addition, they gave a bicriteria, where, for any set of latency functions, the *average latency* in a Nash flow is bounded by twice the *average latency* of an optimal flow with double rates. I adopted their model and checked the behavior of other network parameters. I showed that the *maximum latency* has the same bounds on the ratio as the *average latency* when there is only one pair of source and sink, and that when there are more pairs, the ratio for *maximum latency* can be arbitrary large (depends on $|E|$). For networks with more than one source-sink pair, I suggested future work of comparing the optimal average or maximum *speed*, i.e., when normalizing the latencies with an appropriate distance measure, to that of a Nash flow.

References

[BMW56] M. Beckman, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation.* Yale University Press, 1956.

[KP99] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium in Theoretical Aspects of Computer Science,* pages 404-413, 1999.

[RT00] T. Roughgarden and E. Tardos. How bad is selfish routing? In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science,* pages 93-102, 2000.