Flow Allocation Games: Pricing, Equilibria and Fast Convergence*

John W. Byers† Danny Raz‡

Abstract

We consider the distributed allocation problem of assigning rates to a set of network flows from a new game-theoretic standpoint. Our setting is a fixed capacitated network in which a set of long-lived flows is given and the routes these flows use are pre-ordained and fixed through the duration of the algorithm. Each flow is then given a set of tokens which it may place on the links that it intends to use. The flows then play an iterative game that runs in a sequence of distributed rounds. In each round, the flows place their tokens, then the links provide a tentative rate allocation in proportion to the number of tokens that it receives. Flows are then constrained to a rate that is the minima of the rates allocated to them by their incident links. The flows may then repeatedly reallocate their set of tokens across links to improve their allocation. We characterize the Nash equilibria of this game, relate them to the equilibria in the TCP Vegas analysis, and describe strategies which converge to the equilibria.

1 Introduction

The performance of high speed networks is to a large extent determined by the way the network resources (i.e. bandwidth, buffer space, or processing power) are allocated to the different flows. In ATM networks, this is done explicitly at the switches by the available bit rate (ABR) service class [4], while in the TCP/IP framework the resource allocation is done by a combination of the end to end flow control mechanism of TCP and the queue management policy at the routers.

More broadly, a widely studied class of combinatorial optimization problems seek to optimize the assignment of rates to a set of long-lived network flows in a fixed, capacitated network. In practice, a typical network tends to exhibit complex dynamics that includes the arrival and departure of flows, outages that affect connectivity, and routing changes that impact the routes that flows take through the network. While some of the theoretical

*Invited paper at the 43rd Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, September 2005.

†Computer Science Department, Boston University. Work supported in part by BSF research grant 2000-182 and NSF research grants ANI-9986397 and ANI-0093296. E-mail: `byers@cs.bu.edu`

‡Computer Science Department, The Technion. Work supported in part BSF research grant 2000-182. E-mail: `danny@cs.technion.ac.il`

literature can address these issues in part, much of the classical work focuses on the *static* case, in which the network is fixed, as is the set of flows, and the routes (paths) they use to route data through the network.

Where flow allocation methodologies differ is with regard to the large set of possible objective functions. Researchers and network operators alike could be concerned with issues ranging from fairness to congestion control to aggregate utility maximization. As these diverse objectives may have conflicting priorities, or work at cross-purposes, there is a considerable body of work on each topic, much of which is distinct from the other related work. However, one over-riding theme that is common to these objectives is a focus on obtaining *distributed* solutions, i.e. solutions that can be obtained whereby flows and links operate with only *local information*. A typical definition of local information is that each flow may communicate with its incident links, typically by transmitting control packets along the flow path (or by setting control bits in packet headers), and each link may similarly communicate with its incident flows. While a lengthy discussion of the large body of related work in flow allocation is well beyond the scope of this paper, we briefly recapitulate three main thrusts of research in Section 2: work on defining and achieving fair allocations, work on aggregate utility maximization, and work on developing a rigorous analytical understanding of congestion control algorithms. Each of these tie in to our later findings.

One limitation of the optimization-driven approach discussed above is that it requires careful orchestration and interaction between parties to execute the algorithms correctly. For example, in the context of primal-dual algorithms for either throughput maximization or aggregate utility maximization, convergence is predicated on correct adherence to the algorithms by all flows and all links in the network. In a similar vein, equilibria for TCP Vegas congestion control requires that all endpoints correctly implement (and then use!) the algorithms. Neither of these methods provide strong incentives for users to cooperate, as has been widely documented. For this reason, a game-theoretic approach to rate allocation problems is of interest (see e.g. [6]), especially if the outcomes that it produces either relate closely to or coincide directly with the outcomes of the optimization approaches discussed above. From a practical standpoint, providing a diverse set of mechanisms to achieve quantitatively similar outcomes in the context of rate allocation is highly useful, as a given mechanism may be much more amenable to deployment in a certain network setting, while less helpful in others.

The particular form of our game, which we present formally in Section 4, allows flows to operate autonomously and in their own self-interest, by apportioning a fixed amount of currency across the set of network resources (links) that they intend to consume. On the other hand, the links operate according to a specific set of rules that they must enforce. The key rule in particular that we propose requires links to allocate their resource (bandwidth) in proportion to the amount of currency that flows have invested. Flows are informed of the allocation from each of their incident links and must adhere to a rate that is the minimum of these link allocations. (In the event a flow attempts to exceed its allocated rate, the incident link would drop the excess traffic, affording the flow no advantage in doing so).

We believe our approach has a number of useful properties. First, our game is built on a simple and natural rule that fairly apportions link bandwidth to flows. Second, our game does not require cooperation among flows, as flows that act in their own self-interest can participate while not adversely affecting other users. Third, our game places the simple policing

functionality only on the network entities that are most likely to be able to implement that functionality: on network links and their incident routers. Moreover, the only information needed by these routers is available locally, and no global operation is needed.

Our main contributions in the remainder of the paper are the following: we formalize the game outlined above, define conditions for the Nash equilibria for this game, and demonstrate that an equilibrium allocation always exists. We then draw connections to how our equilibrium allocation compares with the results of other optimization methods. We show a striking similarity between our equilibria and the equilibria that TCP Vegas seeks to achieve, as proven in [10], and also relate our work to max-min fair and throughput-maximizing allocations. Finally, we describe our work in progress on several distributed algorithms for iterative token allocation, including experimental results that show their convergence in practice, as well as a discussion of theoretical running time analysis.

2 Related Work

Early work on optimal resource allocation in high-speed networks focused on global measures such as maximizing the overall throughput of the network, and maximizing the power of the network [5]. Unfortunately, many of these early results were negative, and demonstrated that certain global objectives *could not* be exactly realized with local resource allocation techniques. For that reason, work has moved away from achieving global objectives which cannot be checked locally to objectives which can both be defined and checked locally. The prime example is the rate allocation policy known as *max-min fairness* [3]. In a max-min fair equilibrium, no connection can be allocated a higher rate without reducing the rate of another connection of equal or smaller rate. A max-min fair solution has many desirable fairness properties, as well as the distinction that routers and connections can efficiently check whether the current setting of rates is max-min fair. Of course, the optimization criterion is far removed from that of maximizing resource utilization, and algorithms achieving max-min fairness can and do perform poorly with respect to these measures.

In the last few years several papers question the hegemony of max-min fairness. Kelly [7, 8] studied the connection between the local resource allocation criterion and the global optimized function. He argued that in fact a measure which he calls proportional fairness optimizes a natural global function which is proportional to the sum of a logarithmic utility function of the different flows. Massoulié and Roberts [13] and Kunniyur and Srikant [9] coontinued this line of research and focused on distributed algorithms that can achieve resource allocation using end-to-end mechanisms. It turns out that sliding window AIMD congestion control (like that used in TCP) tends to converge to proportional fairness rather than max-min fairness [8, 11].

A related line of work in the algorithms literature considers fast approximate solutions to positive linear programs, including the seminal work of Plotkin, Shmoys and Tardos [14, 12]. One of the central tenets employed here are primal-dual methods, which were subsequently fitted to distributed approximation algorithms for the problem of flow allocation, first by Awerbuch and Azer [1], and later by Bartal, Byers, and Raz [2], who obtained $(1 + \epsilon)$-approximate convergence in a polylogarithmic number of rounds.

The seminal work of Kelly also focused on primal-dual algorithms for flow control, but

with a more general notion of optimizing concave utility functions [7, 8]. Kelly's notion of interpreting dual variables as link *prices* was widely adopted, and then used to interpret and analyze a variety of window-based congestion controllers, culminating in the full analysis of TCP Vegas [10]. This trend towards an integrated analytical understanding of complex congestion controls has been one of the recent success stories in networking research.

3 Preliminaries and Problem Statement

Before contrasting the pros and cons of particular optimization measures and specific fairness measures, let us revisit the maximization problem we are interested in studying. We assume that we have a set of long-lived connections each of which use a fixed path through the network. (We assume that routes are selected by other network policies and that routes remain stable over large time scales, thus we do not concern ourselves with issues associated with route changes which could occur within the timeframe of a connection). The goal is to assign each of these connections an end-to-end rate so that no edge capacity constraints are violated and so that either a specific function of the rates (such as the sum of the rates) is maximized, or a specific constraints regarding the rates (which reflects fairness criterion) holds throughout the network. In this basic problem, there is no minimum or maximum rate that a given connection may receive, although introduction of these additional side constraints does not fundamentally alter the problem under consideration.

Formally, consider a set of m bandwidth-intensive flows which use fixed-path routes through a network of n links. Each of the links ℓ_i has a capacity c_i, and each flow j uses a path P_j that comprises a subset of the links. An allocation of flow rates F_j is feasible if:

$$\forall i, \sum_{j|i\in P_j} F_j \leq c_i.$$

An important question that arises in this context is: What is strict fairness? As described above max-min fairness is one possible answer, and proportional fairness is another. To understand better the different approaches and their connection to the total amount of flow thought the network let us consider the following example.

Consider the case where m parallel flows pass through k internal network nodes. In any of these nodes, there exists additional l flows that use it (see Figure 1). Assume that the capacity of each node is 1, and the demand of each flow is 1. In the case where $m = l = 1$, we have one long flow with k cross flows. In this case max-min fairness allocation will give half a unit to each of the flows and the overall utilization is $k/2$, while if we want to maximize the throughput we will starve the long flow and allocate one unit to each crossing flow. This results in a total utilization of k, a factor of 2 better than the max-min allocation.

Max-min fairness can provide considerably worse throughput than in this simple case. Let $m = \sqrt{k}-1$, and $l = 1$, then max-min fairness will allocate $\frac{1}{\sqrt{k}}$ unit for each flow, and the total utilization is $\frac{k+\sqrt{k}-1}{\sqrt{k}}$ which is an $O(\sqrt{k})$ factor from the best achievable throughput.

Going back to the problem of fairness definition, the question here is whether we should treat both cross flows and long flows the same way. Note that long flows use more resources per one unit of flow since they go through many more links, and therefore it could be that

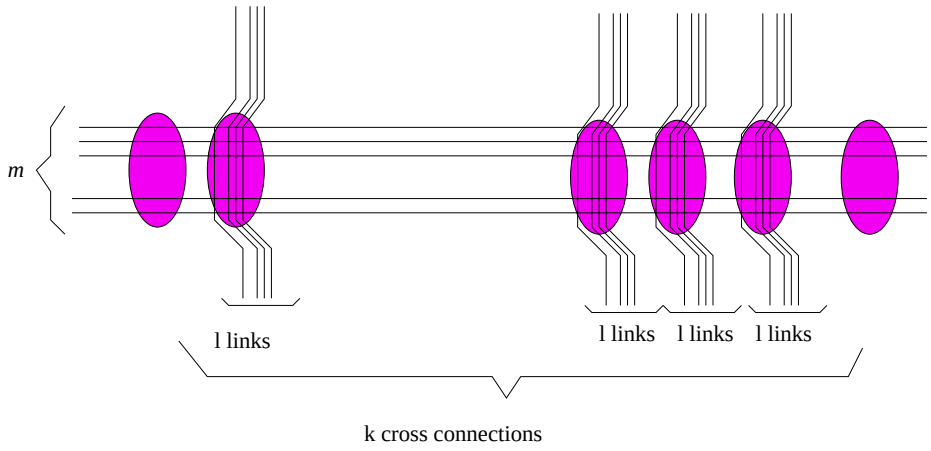

Figure 1: An Example of a Basic Resource Allocation Scenario

in order to be fair these flows should get a smaller allocation of resources at each link. To address this point we introduce a new model in which each flow gets a certain amount of tokens, it can divide these token among all links that it uses, and then the router incident to each link uses some policy based on the number of token put by each flow to decide upon the actual allocation of resources. In this way, if we give all flows the same amount of tokens, we force the network to give more local resources to short flows, and if we decide to give each flow a number of tokens that is proportional to its length, we give the flows an equal opportunity to compete on local resources.

4 Game Formulation

In order to establish a rate allocation, we have flows participate in the following distributed game, which we describe as operating in a sequence of synchronous rounds for the purpose of analysis, but where asynchronous execution is possible. In the first phase of each distributed round, each flow j has a set of T_j tokens from which it may assign a quantity t_{ij} to each link i along P_j, subject to the constraint that for all j, $\sum_i t_{ij} \leq T_j$. We assume that arbitrary fractions of tokens may be placed on any link. We also let w_i denote the total amount of tokens placed by all flows at a link i in a given round, i.e. $w_i = \sum_j t_{ij}$.

In the second phase of each distributed round, each link i then computes a "fair" allocation to each of its participating flows. It does so by computing an allocation f_{ij} that is a weighted proportion of its capacity c_i weighted by the number of placed tokens, if any tokens were placed on it. If no tokens were placed on link i, then the allocation f_{ij} is unrestricted, i.e. $f_{ij} = c_i$. (In the event that a flow tries to game the system by not placing any tokens on a bottleneck link, then it risks getting an allocation of zero once any other flow places tokens on that link). Formally, we have:

$$f_{ij} = \begin{cases} c_i \cdot \frac{t_{ij}}{w_i} & \text{if } w_i > 0 \\ c_i & \text{otherwise} \end{cases}$$

Each flow then receives an allocation F_j equal to the minimum of the allocations provided by the incident links: $F_j = \min_i f_{ij}$. From this discussion, it should be clear that a round proceeds in three steps: the flows autonomously allocate tokens t_{ij}, the links then compute a proportional allocation f_{ij}, then the flows must accept the minimum of the f_{ij}'s they have been allocated. Note that each link i can easily enforce the local allocation of the allocation f_{ij}, and thus no flow j can obtain an end-to-end rate that exceeds F_j.

We now provide some simple facts and definitions, before formulating the equilibrium for this problem in an optimization framework.

Definition 1 *An allocation is* tight *if for all j and for all i such that $t_{ij} > 0$, $f_{ij} = F_j$.*

A tight allocation is one in which every flow has wasted none of the tokens it has placed, i.e. removal of any token from any flow would cause that flow's allocation to be reduced.

Definition 2 *An allocation is* at equilibrium *if it is a tight allocation in which for all j, $\sum_i t_{ij} = T_j$.*

We have an equilibrium allocation when every flow has placed all of its tokens, and has wasted none of the tokens it has placed.

4.1 One interesting example

To exemplify the definitions and to demonstrate certain properties of the game, we consider the following example in the simple case where all links have unit capacity and all flows are allocated the same number of tokens, i.e. $\forall j, T_j = 1$.

Consider a network which is a ring with 5 links labelled 1 through 5, plus an additional link bisecting the ring labelled link 6, and consider the set of six flows using sets of links $\{1,2\}$, $\{2,3\}$, $\{3,4\}$, $\{4,5\}$, $\{5,1\}$ and $\{1,3,6\}$ respectively.

Under the assumption of unit capacities, the max-min fair allocation gives $1/3$ to the five flows which use one of the two links which have three competing connections (i.e. links 1 and 3), and $2/3$ to the flow using the less loaded links $\{4,5\}$ for a total throughput of $7/3$. The throughput-maximizing allocation gives $1/2$ to each of the flows around the ring and nothing to the cross-cutting flow, for a total throughput of $5/2$.

To compute the equilibrium allocation of the game, first note that there is considerable symmetry. The flows using links $\{1,2\}$ and $\{2,3\}$ are equivalent, and will place all of their tokens on the loaded links, and none of their tokens on link 2. The flow using lightly loaded links $\{4,5\}$ will place equal weight on each link. The flow using links $\{1,3,6\}$ will place no weight on link 6, and equal weight on the other two. The remaining two flows must then solve an equivalent problem. Consider the flow using $\{1,5\}$. It must place x units on the heavily loaded link 1 and $1-x$ on the less loaded link 5 to balance the allocation, i.e. to satisfy:

$$\frac{x}{1.5+x} = \frac{1-x}{1/2+1-x},$$

which solves for $x = 3/4$.

The result is that the $\{4,5\}$ flow gets an allocation of $2/3$, the flow using links $\{1,3,6\}$ gets an allocation of $2/9$, the two flows using only one heavy link each get an allocation

of 4/9, and the final two flows where we solved for x get an allocation of 1/3. The total allocation is 22/9. In this case, $7/3 < 22/9 < 5/2$, and in the full version of this manuscript we prove the following more general result.

Theorem 1 *The aggregate throughput, i.e.* $\sum_j F_j$*, in the equilibrium of the game always lies between (but not strictly between) the throughput achieved in a max-min fair allocation and that achieved in a throughput-maximizing allocation.*

5 Equilibria

We start our discussion of the equilibria of this game with two simple observations for which we defer the proofs to the full version.

Claim 1 *If an allocation is* at equilibrium *then for all* j *and for all* $i \in P_j$ *such that* $t_{ij} > 0$

$$F_j = \frac{t_{ij}}{w_i}.$$

Claim 2 *If an allocation is* at equilibrium *then for all* j

$$F_j = \frac{T_j}{\sum_{i \in Pj, t_{ij} > 0} w_i}.$$

We now prove that the flow allocation game described in Section 4 has a Nash equilibrium. To do so, we prove the following theorem.

Theorem 2 *At equilibrium, the allocations in the game solve the following optimization problem:*

$$\max_j T_j \log x_j$$
$$s.t. \forall i, \sum_{j | i \in P_j} x_j \leq c_i$$

Proof: We follow the general line of the proof in the framework of Kelly, Maulloo and Tan [8] to demonstrate convergence properties of primal-dual algorithms for optimizations of this form.

First, consider the Lagrangian relaxation of the problem above with Lagrange multipliers y_ℓ associated with edges of the network. This relaxation has the form

$$L(x, y) = \sum_j T_j \log x_j - \sum_i y_j \left(\sum_j x_j - c_\ell \right).$$

Then, consider the partial derivatives with respect to the primal variables x_j:

$$\frac{\delta L(x, y)}{\delta x_j} = \frac{T_j}{x_j} - \sum_i y_i.$$

Each of these partial derivatives gives a local maximum for x_j when the partial derivative evaluates to zero, i.e. when $x_j = \frac{T_j}{\sum_i y_i}$. If we now interpret the Lagrange multipliers y_i as the total token allocation at link i (that is $y_i = w_i$), then we have that the the balance equation for x_i corresponds to a setting where the tokens are placed in an equilibrium allocation, according to Claim 2. ■

5.1 Comparison to Vegas equilibrium

The conditions for equilibrium are strikingly similar in form to the conditions achieved in the TCP Vegas equilibrium [10]. The primal form of the TCP Vegas optimization is:

$$\max_{s \in X} a_s d_s \log x_s$$
$$\text{s.}t. \sum_{s \in S(\ell)} x_s \leq c_\ell, \ell \in L$$

In this formulation, the d_s are interpreted as round-trip propagation delay values associated with flow s, while a_s reflects a tunable Vegas parameter that sets a threshold for tolerable error between the expected rate and the actual measured rate. When the difference exceeds this a_s threshold, the window size is updated according to the Vegas algorithm. Full details can be found in [10].

6 Algorithms

Given the similarity between the equilibria between the Vegas optimization and that obtained by our game, a first natural consideration is to draw a correspondence between algorithms which obtain convergence for the former and those that do so for the latter. In fact, it is possible to show that an "emulation" of methods which converge to proportional fairness, or more specifically, to the Vegas equilibrium, can be performed in the context of our game. This can be done by using the notions of price and duality of link variables to set the corresponding token allocation at the sender.

However, these methods are not especially fast, and we are interested in deriving provable time bounds on convergence. We currently have several proposed algorithms that converge extremely quickly to a $(1+\epsilon)$-approximation of the Nash equilibrium in experimental settings, but we cannnot yet prove a strong upper bound on their distributed running time. Experimental results are described in more detail in the full version of the manuscript. The fastest of the algorithms we are considering uses the following additional definition. Let g_j be the tokens that are wasted for connection j in a given round, i.e. those that could have been removed without changing the value of F_j.

For the $1 - g_j$ tokens that flow j did not waste, it received a flow value of F_j. If it had allocated the remaining g_j tokens usefully and in proportion to the ones it did not waste, it would have received a flow value of of $F_j/(1 - g_j)$. So we can compute a notion of wasted flow:

$$z_j = F_j - \frac{F_j}{1 - g_j} = \frac{F_j(1 - g_j) - F_j}{1 - g_j} = \frac{F_j g_j}{1 - g_j}.$$

Here is a summary of the algorithm:

1. At each round, for each flow j, flow j places a token allocation t_{ij}.
2. Each link divides each t_{ij} by w_i to arrive at a tentative flow value f_{ij}.
3. The flow receives $F_j = \min_i f_{ij}$, removes its garbage tokens g_j and computes its wasted flow z_j.
4. To compute t_{ij} in the next round, it scales up the non-wasted tokens by a factor of $\frac{1}{1-g_j}$.

From this definition, we have that,

$$\hat{t_{ij}} = \begin{cases} \left(1 + \frac{g_j}{1-g_j}\right) t_{ij} & \text{if } f_{ij} = f_i \\ \left(1 + \frac{g_j}{1-g_j}\right) F_j w_i & \text{otherwise} \end{cases}$$

In the above description, it is also noteworthy that we can count the waste in two different ways. We have counted the tokens lost on a per-flow basis, but we can also count the tokens lost on a per-router basis. Define R_i to be $\sum_j f_{ij} - F_j$. We can also define a router's wasted capacity (analogous to wasted flow) by R_i/W_i. It follows immediately that $\sum_i R_i = \sum_j g_j$. Our current line of attack attempts to define appropriate potential functions related to these quantities, although identifying a quantity that is provably monotone through the course of the algorithm appears difficult.

7 Conclusion

In the natural problem domain of deciding upon a distributed allocation of rates to long-lived flows in a network, among the first questions to consider is to determine what to optimize. But an equally natural question is to consider whether the optimization of interest is also in line with the presumptive self-interest of individual connections. Our main contribution is a new class of distributed games that model this flow allocation problem in a natural way that does not necessitate cooperation among flows, and places the policing functionality at network routers. We demonstrate that our game has a Nash equilibrium that is a variant of proportional fairness that penalizes flows traversing more bottleneck links, and discuss our ongoing work to develop fast algorithms to converge to these equilibria.

Acknowledgments

We thank Ashish Goel and Ramesh Johari for helpful conversations, and in particular for making the connection between the equilibria in our game and the equilibria of the Kelly framework, leading to their original proof sketch of Theorem 1.

References

[1] B. Awerbuch and Y. Azar. Local optimization of global objectives: Competitive distributed deadlock resolution and resource allocation. In *Proc. of the 35th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 240–249, 1994.

[2] Y. Bartal, J. W. Byers, and D. Raz. Fast, distributed approximation algorithms for positive linear programming with applications to flow control. *Siam J. Computing*, 33(6):1261–1279, August 2004. Preliminary version appeared in *IEEE FOCS '97*.

[3] D. Bertsekas and R. Gallagher. *Data Networks*. Prentice Hall, 1987.

[4] S Chong, E. J. Hernandez-Valencia, L. Benmohamed, and R. Nagarajan. Rate control for ATM ABR service. *Europ. Trans. Telecom.*, 8:7–20, 1997.

[5] J. Jaffe. Bottleneck flow control. *IEEE Transactions on Communication*, 1(7):954–962, 1981.

[6] R. Johari and J. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.

[7] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, January 1997.

[8] F. P. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *J. Operations Research Society*, 49(3):237–252, March 2002.

[9] S. Kunniyur and R. Srikant. End-to-end congestion control schemes: Utility functions, random losses and ECN marks. In *Proc. of IEEE Infocom 2000*.

[10] S. Low, L. Peterson, and L. Wang. Understanding TCP Vegas: A duality model. *Journal of the ACM*, 49(2):207–235, March 2002.

[11] S. H. Low and D. E. Lapsley. Optimization flow control I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, December 1999.

[12] M. Luby and N. Nisan. A parallel approximation algorithm for positive linear programming. In *Proc. of 25th ACM Symposium on Theory of Computing*, pages 448–457, 1993.

[13] L. Massoulié and J. Roberts. Bandwidth sharing: objectives and algorithms. In *IEEE Infocom 1999*, pages 1395 – 1403, 1999.

[14] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. In *Proc. of the 36th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1995.