From *SIAM News*, Volume 38, Number 3, April 2005

Computer Scientists Optimize Innovative Ad Auction

By Sara Robinson

As one of the few Internet companies to make a mathematical idea pay off, Google has long been a darling of the research community. PageRank, the company's sophisticated algorithm for ranking search results, is still a focus of mathematical research (see "The Ongoing Search for Efficient Web Search Algorithms," *SIAM News*, November 2004, http://www.siam.org/siamnews/11-04/websearch.pdf). But a clever search algorithm alone did not create a multi-billion-dollar company. The key to Google's financial success is an innovative auction system for selling the advertisements that appear alongside search results.

Google's auction advertising model, too, is now proving to be a source of interesting mathematical research problems. Recently, four computer scientists—Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani—devised an algorithm for optimally ranking advertisements.

The new algorithm is a generalization of a well-known theoretical result published 14 years earlier. "We were lucky," Saberi says. "It's a cute problem that also has a mathematically interesting answer."

The World's Biggest Auction House

Of the many search sites launched in the high-tech frenzy of the late 1990s, only Google, which went online in 1998, had an effective way of determining the most relevant sites for a given query. Whereas other search engines simply counted frequency and location of keywords, Google found a way to rank all Web sites in order of their popularity as indicated by the link topology of the Web, and used this master ranking list as a guide in determining relevancy. PageRank's effectiveness enabled Google to quickly snag a major share of the search market.

But Google's revenue came from advertising, not searches, and initially, the Internet advertising market was small. No one had figured out how to exploit the unique features of the Internet for advertising, and traditional media, such as television and magazines, still attracted the lion's share of company advertising budgets.

A single clever idea, however, has turned the Internet, and, in particular, search engines, into a mecca for advertisers. By linking advertisements to search terms, search engines can provide advertisers with precisely targeted audiences. Someone who has just searched for "wedding dresses," for example, is more likely to respond to an advertisement for a wedding planner or honeymoon package.

The germ of the idea came in 1997 from an Internet start-up called Overture, which was eventually acquired by Yahoo. Overture's idea was to rank search results by auction, giving top billing to the highest bidder. The company's executives argued that this method would provide good search results as well as revenue, because sites would bid large amounts only for search terms relevant to their content.

Google's founders, who, to their credit, have always strived to separate advertising from search rankings, apparently were not impressed with this argument. In 2000, the company introduced its own variant of Overture's model. Rather than auctioning off search rankings, Google established an auction only for advertisements, and separated them from the search results. Initially, advertisers paid for each "impression," that is, each time an ad and associated link were displayed after a search. Now, an advertiser pays only when someone clicks on the link.

In the current Overture/Yahoo model, which, like Google, separates ads from search results, advertisers choose one or more search keywords to which they wish to link their ads and bid what they are willing to pay each time someone clicks on the link. The ad links appear in a separate list next to the search results for those keywords, ranked in order of the bids.

Google's model is similar but differs in some key respects. Because unappealing or poorly targeted ad links attract relatively few clicks and thus provide less revenue for Google, the company takes into account each ad's "clickthrough" rate—the probability that the searcher will click on the ad link—as well as the bid. Unlike Yahoo/Overture users, Google advertisers cannot see the bids of their competitors, but the amount they actually pay per click is the minimum needed to rank above the next highest bidder. Another feature unique to Google, and key to the new ad-ranking algorithm, is that advertisers can specify a daily budget. Once the budgeted amount is spent, the ad is dropped from the list.

Bids for keywords can range from as little as a few cents a click, for obscure search terms like the word "albeit," to nearly $100 per click, for search terms that, like Vioxx, are attached to lucrative lawsuits. Because the entire system is constantly in flux, there is a new auction each time someone performs a search on Google. With millions of queries a day, Google has become the world's largest auction house, and Yahoo is just behind.

A Thought-Provoking Job Interview

In March of last year, Amin Saberi, then finishing a PhD in theoretical computer science at the Georgia Institute of Technology, visited Google to interview for a job. Though he eventually landed elsewhere, the visit proved fruitful: Monika Henzinger, then the company's director of research, posed an intriguing question.

At the time, Google was ranking ads by the product of the clickthrough rate and the bid. Henzinger wondered if it was possible to improve this ranking algorithm and quantify its effectiveness.

Back at Georgia Tech, Saberi relayed the question to Mehta, a fellow graduate student, and Vijay Vazirani, their thesis adviser. After some discussion, it seemed to the researchers that the advertisers' daily budgets should play a role in the algorithm. It is in Google's interest to ensure that each advertiser remains solvent as long as possible; thus, the researchers concluded, the algorithm should somehow weigh the remaining fraction of each advertiser's budget against the amount of its bid.

This conclusion defined a research problem: Over the course of a day, what strategy should Google use to rank the ads for each query so that the advertisers spend the greatest possible fraction of their aggregate daily budget?

Over the course of a day, what strategy should Google use to rank the ads for each query so that the advertisers spend the greatest possible fraction of their aggregate daily budget?

A key aspect of the problem is the environment of imperfect information. Because a given day's search queries cannot be determined in advance, a ranking algorithm has to rank the ads as the queries come in, based on the bids and remaining fractions of the advertisers' budgets. Such algorithms, which operate in real time without access to the entire data set, are known as "online algorithms." Their effectiveness is evaluated as a "competitive ratio," which compares the worst-case output of the algorithm to the ideal solution.

Vazirani immediately realized that a basis for a solution lay in what is known as the online bipartite matching problem, an optimal solution to which he had provided 14 years earlier, with his brother, Umesh Vazirani, and Richard Karp, both professors of computer science at the University of California, Berkeley. "We saw online matching as a beautiful research problem with purely theoretical appeal," Umesh Vazirani says. "At the time, we had no idea it would turn out to have practical value."

The online matching problem is this:

A matchmaker and n boys are gathered in a room, in which n girls appear, one at a time. Each girl has a list of boys who are acceptable to her, which she reveals to the matchmaker as she appears. The matchmaker matches her to one of the boys on her list or, if they are all taken, leaves her unmatched. The matchmaker's goal is to maximize the total number of matches.

It's easy to see that no deterministic matchmaker algorithm can guarantee a competitive ratio of better than ½. The Vaziranis and Karp thus tried a simple randomized approach in which each girl, on arriving in the room, randomly chooses from among the boys acceptable to her. When they tried to show that this approach is optimal, they came up with the following worst-case example:

Imagine 100 boys and 100 girls, each group numbered from 1 to 100, and suppose that each girl is willing to match to the boy with the same number, and that each of the first 50 girls also likes each of the last 50 boys. In other words, each of the first 50 girls likes 51 boys, while each of the last 50 likes only one. If each girl appears in numerical order, and rolls a die to determine which of the acceptable boys she will match with, with high probability, each of the first 50 girls will match to one of the last 50 boys, leaving most of the later girls unmatched. Thus, the simple randomized algorithm, too, achieves only ½ the optimal allocation, which is to match each boy to the girl with the same number.

With this example in mind, the researchers thought about a different approach: What if they fixed a single random permutation of the boys in advance, and had the matchmaker match each girl to the highest ranked (in the permutation) of the remaining acceptable boys? Thus, instead of rolling a die each time a girl appeared, the matchmaker would roll the die only once, at the beginning of the algorithm. For the example above, they realized, this matchmaking algorithm performs far better: Some of the boys numbered 1–50 would be likely to come up early in the permutation, enabling them to be matched to the girls with the same numbers. The researchers were able to show that this approach is optimal in general, achieving a competitive ratio of $1 - 1/e$.

The online matching problem is the special case of the Google problem in which advertisers are restricted to unit bids and unit daily budgets, and a single ad is allocated to each query. The case with larger budgets was also addressed, in a theoretical computer science research paper published in 2000. In place of the boys, the researchers—Bala Kalyanasundaram and Kirk Pruhs—envisioned a variety of items, each costing $1. The girls are replaced by buyers, each with a budget of b dollars to spend and a desire for only some of the items. The problem, which the researchers called b-matching, is to allocate items to buyers in real time, in a way that minimizes the aggregate remaining budget. Kalyanasundaram and Pruhs's algorithm simply assigns each item to the buyer with the largest remaining budget. This approach also achieves a competitive ratio of $1 - 1/e$, which the researchers showed to be optimal for a deterministic algorithm.

To solve the Google problem, the Georgia Tech researchers realized that they needed to generalize online matching and b-matching to the case of varying budgets and bids. In the Google setting, the boys or items become the search queries typed into Google in the course of a day, and the girls or buyers are the advertisers, with their bids and daily budgets. The problem is to use the budgets and bids to determine the allocation of advertisers to queries.

Initially, the researchers simplified the problem, assuming that only one ad is allocated to each query. Later, they generalized their solution to the case of ranking multiple ads for each query.

The Optimal Tradeoff

Over the next few months, Mehta, Saberi, and Vijay Vazirani worked on the Google problem, later joined by Umesh Vazirani. Eventually, they solved it by providing a function Ψ that gives the optimal tradeoff between bid and remaining budget for determining query allocations. The resulting algorithm, like online and b-matching, achieves a competitive ratio of $1 - 1/e$. To prove their result, the researchers introduce a new mathematical tool—a "tradeoff-revealing linear program"—and give simpler proofs of optimality for the online and b-matching algorithms.

In searching for an approach to the Google problem, Vijay Vazirani turned to the tools of his specialty: approximation algorithms. Approximation algorithms yield near-optimal approximate solutions to (typically NP-hard) optimization problems, and are evaluated by the worst-case ratio of the cost of the approximate solution to that of the optimal one.

A couple of years earlier, while trying to determine the approximation factor (the worst-case ratio) for another algorithm, Vijay Vazirani, Saberi, and three colleagues had come up with a linear programming trick that has proved to be a useful tool for other problems as well. For a problem of size n, they had written down the constraints satisfied by the solution, and then let the ratio of the approximate solution to the optimal one be the objective function of a linear program. They were able to find a generalized solution to the resulting sequence of linear programs, which gave the worst-case performance of the algorithm. They called their technique a "factor-revealing" LP, because it reveals the approximation factor for the algorithm.

Mehta and Saberi were able to apply the factor-revealing LP technique to the b-matching problem, giving a simpler proof that the competitive ratio for the Kalyanasundaram–Pruhs algorithm is $1 - 1/e$. The Google problem, however, was far more complicated. It wasn't clear how they could write constraints for a linear program without knowing the correct tradeoff function, and even if they fixed a tradeoff function, the constraints would vary depending on the day's sequence of queries.

Nevertheless, the four researchers went ahead and wrote down what looked to them like a horrendous mess: For each n, they had a linear program that had the same variables and objective function as the factor-revealing LP. The constraints, too, looked similar, except that the right-hand sides of the inequalities depended on the unknown tradeoff function, Ψ, and query sequence, π.

Surprisingly, although this linear program appears to carry no real information, its dual program provides a great deal of valuable information. Because the unknown quantities in the primal LP appear only in the right-hand sides of the linear constraints, they affect only the objective function of the dual. Thus, the actual convex polytope over which the optimization is carried out is the same regardless of the tradeoff function Ψ or query sequence π.

Remarkably, the researchers eventually showed, the optimal value of the dual is attained at a fixed vertex of the polytope, regardless of Ψ or π. This fact enabled them to select a tradeoff function that achieves the same $1 - 1/e$ bound on the competitive ratio as b-matching, which Kalyanasundaram and Pruhs had shown to be optimal for deterministic algorithms. The researchers call this new technique a "tradeoff-revealing" LP. "At a deep level, we don't really understand this technique," says Umesh Vazirani. "It works like magic."

The optimal tradeoff function is $\Psi(x) = 1 - e^{-x}$. Google, then, will maximize its profits if it allocates queries to the advertiser with the highest value of $d\,\Psi(f)$, where d is the bid and f is the fraction of the budget that has not yet been spent.

Once they had the tradeoff function, the researchers were able to give an alternative randomized algorithm that achieves the same competitive ratio, $1 - 1/e$, which they also showed to be optimal for randomized algorithms. As with the online matching problem, the randomized algorithm chooses a random permutation of the advertisers in advance, and ranks the ads in order of $d\,\Psi$ (rank/n), where n is the total number of advertisers and rank is the advertiser's rank in the permutation. To prove that this algorithm also achieves a competitive ratio of $1 - 1/e$, they used the techniques from the original online matching paper.

Curiously, these two algorithms achieve the same effect, although one takes budgets into account and the other does not. "Randomization is an amazing tool," says Vijay Vazirani. "Even though the randomized version of the algorithm doesn't take budgets into account, it foils the worst-case query sequence."

For Google, the randomized algorithm may have some practical advantages, the researchers note in their preliminary writeup. In the deterministic algorithm, each server providing Google search results must keep track of all the advertisers' budgets at each point in time. With the randomized algorithm, the server has to know only that day's permutation of the advertisers.

Translating Theory into Practice

The researchers have not yet published their work, but Saberi, now at Stanford, gave a talk on the algorithm that some Google researchers attended. He was subsequently invited to give another talk on the result at Google. Henzinger, who originally posed the problem to Saberi, has since taken an academic post at Ecole Polytechnique Fédérale de Lausanne but continues to consult for Google. She requested a video of Saberi's talk.

The researchers intend to apply for a patent, but only to ensure that the research remains in the public domain, where they and others can continue to build on it. One direction they may explore, says Vijay Vazirani, is how an algorithm might exploit statistical data about Google's search queries. The current algorithm assumes that Google has no advance information about each day's sequence of queries when, in fact, it does.

Another possible research direction is the game theoretic angle. Ideally, says Mehta, who plans to graduate in June, a ranking algorithm should minimize manipulation of the market by encouraging advertisers to reveal their true valuations for a given keyword and their true daily budgets.

A third direction, and Mehta's current focus, is to look for new applications for tradeoff-revealing LPs. Tradeoffs come up all the time in engineering problems, he notes, and occasionally in computer science. "We are providing a formal method of finding an optimal tradeoff function," he says. "I hope to show that this is not a one-off thing, but a useful technology that will apply to other problems as well."

Sara Robinson is a freelance writer based in Los Angeles, California.