

The

HONEY BEE

Algorithm:

A Biologically Inspired Approach to Internet Server Optimization

By Dr. Craig A. Tovey

DR. CRAIG A. TOVEY is a professor in the College of Computing and the School of Industrial and Systems Engineering at Georgia Tech. His principal research and teaching activities are in optimization, probabilistic analysis, and natural systems. His current research concerns the effectiveness of valid inequalities in integer programming, classical and biomimetic algorithms for robots and web-hosting, the formation of social dominance hierarchy structures, and sustainability measurement. The following article details one of his current research projects in natural systems.

When Sunil Nakrani knocked on the door, ISyE Professor Craig Tovey didn't know that a 15-year old dream of his was going to be realized. Nakrani, a visiting scholar in the School of Electrical and Computer Engineering at Georgia Tech and a doctoral student at the University of Oxford, was searching for a method to allocate computers among different clients at a web-hosting facility. He approached Tovey based on the professor's reputation for algorithm heuristics. He did not know that Tovey had studied how honey bee colonies allocate foragers among different flower patches and had been searching for an industrial application of the bees' method. But that all changed as a result of this meeting. The two researchers began working together to model the dynamic server allocation problem and proposed a biologically inspired approach to the optimization problem in a managed Internet server colony. Within the past two years, Nakrani and Tovey have developed an algorithm that mimics the behavior of honey bee foragers, with very promising test results for a simulated web-host center.

In ISyE, Tovey explains, "we learn about natural systems, and we learn from natural systems. This project has some of both." Years ago, Professors John Bartholdi, John H. Vande Vate, and Tovey used Operations Research (OR) techniques to help explain how bees allocate foragers among flower patches in order to bring a lot of nectar into the hive. Now, Nakrani and Tovey have imitated the bees to allocate computers among web clients in an effort to bring a lot of money into a web-host center. And the test results give new insight as to why the bees' strategy helps them survive.

To provide some background, Nakrani first encountered the web-host allocation problem while working at IBM. Web-hosting is now a \$30 billion/year and growing industry. Every time you check the weather, buy merchandise, or pay your bills online, chances are the computer you connect to is not actually run by the weather station, retail store, or bank. Instead, you most likely connect to a computer at a facility which runs several web-applications, or web-apps for short, on a large bank of computers. By aggregating the different and highly variable demand patterns of its various web-

app clients, the hosting center can achieve an economy of scale, as shown in Figure 1. (The same computer which provides weather reports in the morning can serve shoppers in the evening.) Web-app clients pay a small fee for each customer serviced, but for security reasons, only one web-app may be loaded onto a computer at a time. Therefore in order to maximize its revenue, the web-host facility must decide how many computers will serve each web-app, while adapting to the changing levels of customer demand.

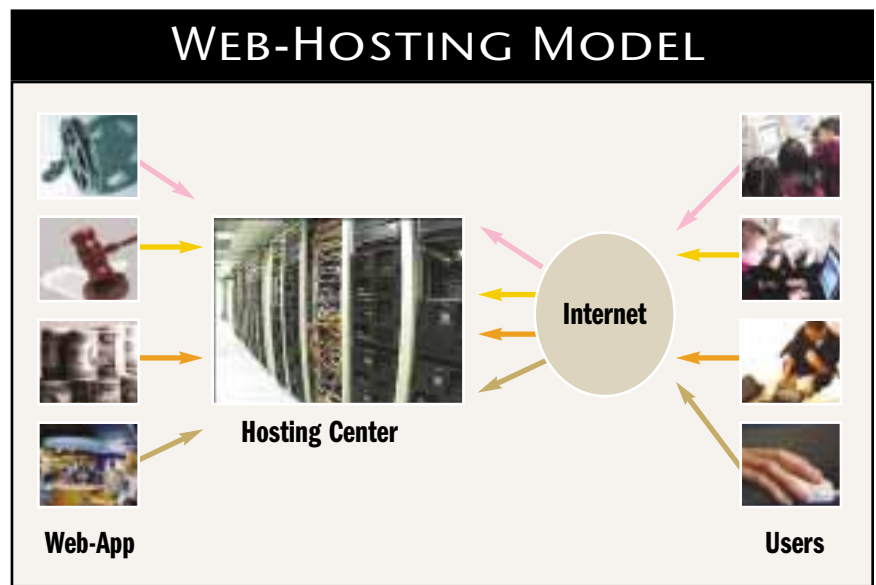


Figure 1

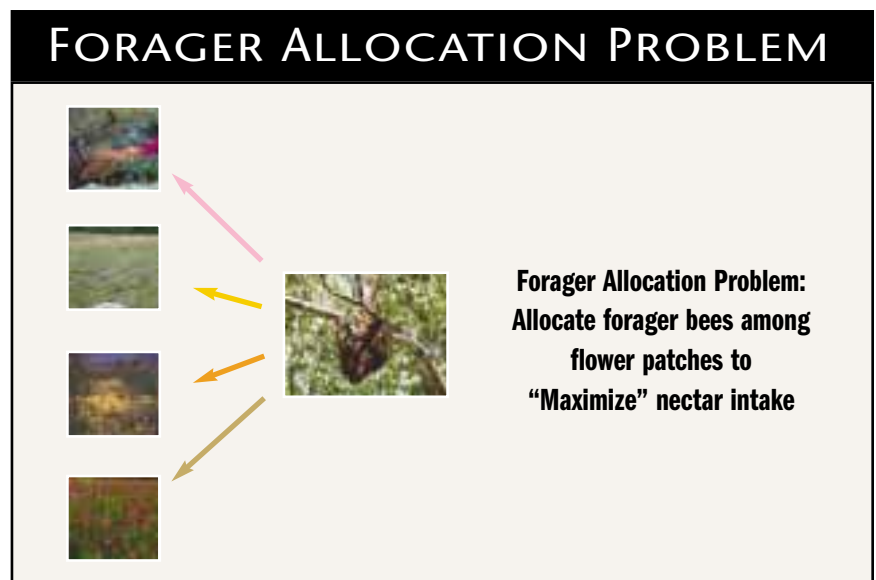


Figure 2

In turn, honey bee colonies operate using the same basic principle. Each colony must collect extra nectar during the warm season to make and store enough honey – usually 20 to 50 kg – in order to survive the winter. Efficient nectar collection is thus crucial to colony survival. It is inefficient, in general, for all of the colony’s foragers to collect from the same flower patch. A large number of bees at one patch can “swamp out” the flowers’ capacity to generate nectar. On the other hand, some flower patches are richer or more productive than others. To maximize nectar intake, the honey bee colony must “decide” in some decentralized but intelligent fashion how many bees will forage at each flower patch. (Figure 2).

As Nakrani described the web-host problem to Tovey, the apparent resemblance to the honey bee problem was obvious. In fact as he moved to the next level of detail, Tovey says, “I got very excited about the potential, because at the deeper level, the problems continued to match up beautifully.” For example, when a computer is switched from one web-app to another, it incurs a 5 to 7 minute downtime (to be scrubbed and reconfigured) for security reasons. A honey bee who is switching patches usually requires several attempts to successfully find the new patch, incurring a similar downtime in effect. As another example, the time to serve a web-app customer has both a fixed-cost component, depending on the web-app, and a variable cost, which increases as the number of computers assigned to the web-app increases. Similarly, the time a forager requires to collect a stomach-full of nectar has a fixed-cost component, the round-trip travel time, depending on the patch location, and a variable cost, the collection time, which increases as the larger number of assigned bees swamps out the patch’s nectar production. After these and other similarities revealed themselves, Tovey presumed that he had finally found the appropriate application of the honey bee method.

“I got very excited about the potential, because at the deeper level, the problems continued to match up beautifully.”


First, the researchers devised a careful test of the honey bee method. They implemented it and three other algorithms in order to assess performance. One of the other algorithms was a standard myopic/greedy algorithm, a traditional optimization method. The second was an optimal static algorithm, which provides an upper bound on current-day practice at many facilities, which only reset their allocation once a month. The third calculated a theoretical upper bound on revenue

Like many biologically inspired heuristics, the method has advantages over many other types of algorithms.

collection, as if the web host center knew the future customer arrival data in advance. They ran a battery of tests on both real trace data from a service provider (NDA) and simulated web traffic.

As the tests began, the researchers were hoping that the honey bee method would be on par with traditional methods and current usage. Like many biologically inspired heuristics, the method has advantages over many other types of algorithms. It is simple, less centralized, and robust under even undetected breakdowns. If its revenue performance were competitive, these other features would give it an edge. However, the results were better than Nakrani and Tovey had hoped. The honey bee method earned more revenue than either the greedy or static methods over a range of data parameters.

When web traffic was not very variable (real web traffic is), the honey bee heuristic no longer performed so well. This gave a new insight into the bees themselves. Earlier work of Bartholdi, Vande Vate, Tovey and Seeley (building on work of von Frisch, Seeley, and others) showed what the allocation pattern is, and that while not optimal it is a good heuristic method. It is not optimal in the sense that if you look at the bees at any one particular time, a different allocation would get more nectar, given the way things are at the present moment. A different way to say this is that if the environment were static, unchanging, then the bees’ allocation pattern does not maximize nectar influx to the hive.

The reason is that to achieve static optimality, the derivatives, i.e. the marginal revenue at each patch, must be equalized. If one extra bee is more valuable at patch A than at patch B, then it would be better to switch a bee from B to A. The honey bee colony does not do that. Instead, the colony equalizes the average return per bee. Nakrani and Tovey suggest that the colony would have to switch allocations one bee at a time, in order to be able to manifest the value of the derivative of nectar influx. Instead, the colony permits many bees to switch at the same time, but it can only manifest the value of the average nectar influx. The colony seems to have traded off optimality in-the-moment for rapid adaptability to change. This may be a very good lesson for human optimizers. 

For additional information about the honey bee algorithm, please contact Dr. Craig Tovey at craig.tovey@isye.gatech.edu.