NEGOTIATING FOR GOODS AND SERVICES

Michael N. Huhns • University of South Carolina • huhns@sc.edu
Anuj K. Malhotra • University of South Carolina • anuj@sc.edu

In 1789, mutinous seamen aboard the *H.M.S. Bounty* cast Captain Bligh and his crew of 17 adrift in an open boat.[1] After sailing many days without food, the desperate men managed to snare a small bird. How could they share it? Captain Bligh wanted the portions, although meager, to be given out fairly. He carefully cut the bird into 18 equal-sized pieces. Two crewmen then engaged in the following protocol: the first chose a portion randomly and called out, "Who shall have this?" The second, with his back to the proceedings, called out a crewman's name.

Although fair, this negotiation protocol for distributing resources did not lead to an envy-free distribution, because some pieces, such as the feet or beak, were clearly less desirable. Can a protocol be both fair and envy-free? Under certain circumstances, yes. For example, the cake-cutting protocol, which is at least 2,800 years old, is both fair and envy-free for two people.[2] One person divides a cake in two, and the other chooses the first piece. This is fair in that each believes he or she received at least half, and envy-free in that neither would wish to trade. This protocol succeeds even when the participants disagree about the portions' value: while one might simply want the largest possible piece, the other might prefer a smaller piece because it has more frosting.

Letting Agents Cut the Cake

Since computational agents don't eat birds or cake, how will such protocols matter to them and when will they apply? We will show how protocols can provide possibilities for benevolence and prove useful in agent-based auctions, the subjects of our two most recent columns.[3,4]

First, agents can use the cake-cutting protocol when negotiating for resources. For example, two agents sharing a processor might each require CPU time and might also wish to complete their computations as soon as possible. One agent might select the sizes of the processor's time slices, while the other would have first choice of the resulting slices. Similar negotiations could occur over bandwidth, relative cache sizes, locks on databases, storage space in a file system, or task decomposition and distribution.

Advanced Cake Cutting

What if there are more than two agents? Envy-free apportioning is more difficult than fair apportioning because an envy-free division is always fair, while in a fair division someone may still end up with the feet or beak. So we will consider fair apportioning first.

One fair protocol for N agents involves the agents' successively reducing a slice until they deem it fair—having a value of $1/N$.[5,6] The last agent to reduce the slice must accept it. The rest of the agents then repeat this procedure for the remainder of whatever they are dividing. This trimming protocol is fair because each agent believes its portion is at least $1/N$ of the total, but it is not envy-free, because one agent might believe another has received a larger share.

The moving-knife protocol, mediated by an auctioneer, is similar and proceeds as follows: the auctioneer slowly increases the size of a time slice (or other desired resource) until one agent assesses the value at $1/N$ and yells, "Stop!" That agent is awarded the slice. The auctioneer repeats the protocol for the rest of the $N - 1$ agents and the remainder of the resource. The last agent might end up with a larger or smaller piece than everyone else, so this protocol is not envy-free.

In another fair protocol, two of the contending agents would first divide a resource equally. A third agent would then negotiate with each of them to obtain one third of each one's half. A fourth agent would then negotiate with the other three agents to obtain one fourth of each one's third. This successive-pairs protocol continues until all N agents have negotiated for their portions.

Divide-and-conquer, another fair protocol, requires the division of a resource to a point where it could be easily divided among two or three agents. According to this technique, one of the agents divides the resource first into halves, and the number of contending agents is also divided into halves. The agents in each half should be happy to get a fair share of their half of the resource. These halves and the agents are again divided in half, and so on until the remaining pieces of the resource can be divided among just two or three agents each.

An Envy-Free Protocol

A simple envy-free protocol for N agents, which unfortunately works only in restricted domains, proceeds as follows: assume that three agents wish to reserve time during a fixed interval. Each agent would like to have as much time as possible and also the earliest time possible.

First, each agent is asked to divide the interval into three pieces that it thinks are fair and that it would be willing to accept. If the agents all value each piece equally, then they would divide the interval in the same way and the pieces could be assigned to them arbitrarily. Each would be happy with its piece.

If the agents value pieces differently, then each would divide the interval differently, as shown in Figure 1. In this example, the green agent would be assigned its leftmost piece (from the start to G1), the blue agent would be assigned its rightmost piece (from B2 to the end), and the red agent would be assigned its middle piece (from R1 to R2). Each agent would have one of the pieces it thought was fair, and the other agents (in its estimation) would have smaller pieces. No agent would be envious. Amazingly, part of the interval would be left over (from G1 to R1 and from R2 to B2). This protocol works for any domain that can be linearized and for any number of agents.

Requirements and Properties

Good negotiation protocols should offer more than just fair or envy-free distributions. They should be well defined and readily available so that agents can easily implement and use them. It seems reasonable to base them on speech acts and incorporate them into standard agent communication languages such as KQML and FIPA's ACL. Adopting a language with precise semantics will make it easier for agents to use the protocols without misunderstandings.

Negotiation protocols should also be simple and efficient to implement, stable (so that no agent has an incentive to deviate from the protocol), distributed (so that no central decision-maker is needed), and symmetric (not inappropriately biased against any agent).[7] Furthermore, the protocols should lead to optimal solutions.

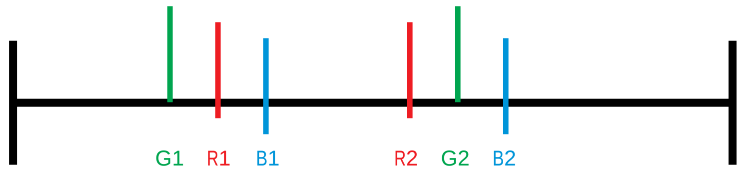

Figure 1. Red, green, and blue agents divide the interval into what each sees as three fair pieces.

System of the Bimonth

To create agents that will negotiate on your behalf, visit the Bazaar at MIT's Market Maker Testing Site. Check it out!

To cut cakes (or allocate chores or determine fair rent obligations among roommates) for any number of people, try the online Fair Division Calculator at Harvey Mudd College.

The list below includes additional Web addresses for other activities related to agent negotiation.

Bazaar • maker.media.mit.edu/
Fair Division Calculator • www.math.hmc.edu/~su/fairdivision/calc/
Kasbah • ecommerce.media.mit.edu/Kasbah/screemshotsindex.html
Agent-Mediated Integrative Negotiation for Electronic Commerce • guttman.www.media.mit.edu/people/guttman/research/commerce/talk10/sld001.htm

An efficient algorithm for cake cutting, such as the auctioneer example, requires just $N - 1$ cuts—the minimum possible number of cuts required to divide the resource among N agents. The trimming protocol would require $N(N - 1)/2$ cuts, and the successive-pairs protocol would use $N! - 1$ cuts. The divide-and-conquer protocol would require approximately $N \log 2N$ cuts, the minimum number needed for performing the division without auctioning.

Negotiating about Tasks

Most agent-based negotiation protocols were derived from results in game theory and serve to allocate tasks.[7] For example, two agents might have to visit a number of Web sites to search for information. Rather than both visiting the same sites, they could exhibit benevolence by agreeing to exchange some of their search tasks. What strategy and protocol should they use to minimize their tasks?

With the monotonic concession protocol, the agents successively compromise their positions by agreeing to accept additional tasks. Negotiation ends when neither agrees to compromise further.

A Zeuthen strategy, in which each agent must be aware of how the other values each task, requires the agent having an advantage to make a concession. This process repeats until the agents reach agreement. The result is an optimum task allocation.

Agent-Based Web Auctions

The two most important requirements for online auctions are that they be fair and secure.[4] They should also introduce low overhead in terms of the cost they add to each transaction and the complexity they introduce into the process. Auctions can be used not only for goods and services, but also for information.

The Dutch auction is one of several auction types developed to meet various requirements. In a Dutch auction, the price descends from a sufficiently high number until one participant elects to buy at the current price.

The Dutch auction closely resembles the moving-knife protocol for resource allocation and could be extended to N agents as follows. An auctioneer begins with a certain price for a portion of a resource and steadily reduces the asking price until one agent accepts. The agent pays whatever price has been agreed on and relinquishes its claim on the rest of the resource. Then a similar auction follows for the remaining $N - 1$ agents and resources. The process repeats until all the resources have been allocated. Such negotiation protocols could serve to auction any resources that can be fairly distributed in the dimensions of time or space—for example, allocations of storage, CPU time, and search results.

Bottom Line

So, how does your organization manage its resources? Does a company vice president using the Web to do market research for an upcoming product rollout or an IPO get network priority over a clerk's search for the best price on a Goo Goo Dolls CD? For most organizations, the answer is no. If your organization isn't using negotiating agents to manage its critical resources, shouldn't it be? ■

REFERENCES

1. C. Nordhoff and J.N. Hall, *Men Against the Sea*, Little, Brown and Co., Boston, Mass., 1933.
2. J. Robertson and W. Webb, *Cake Cutting Algorithms*, A.K. Peters, Natick, Mass., 1998.
3. M.N. Huhns and A.M. Mohamed, "Benevolent Agents," *IEEE Internet Computing*, Vol. 3, No. 2, Mar.-Apr. 1999, pp. 96-98.
4. M.N. Huhns and J.M. Vidal, "Online Auctions," *IEEE Internet Computing*, Vol. 3, No. 3, May-June 1999, pp. 103-105.
5. I. Stewart, "Mathematical Recreations: Your Half's Bigger Than My Half!" *Scientific American*, Dec. 1998, pp. 112-114.
6. I. Stewart, "Mathematical Recreations: Division without Envy," *Scientific American*, Jan. 1999, pp. 110-111.
7. J.S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*, MIT Press, Cambridge, Mass., 1994.

Michael N. Huhns is a professor of electrical and computer engineering at the University of South Carolina.

Anuj K. Malhotra is a graduate student in the Department of Electrical and Computer Engineering at the University of South Carolina. His current research interests are multiagent systems and distributed databases. He earned a BE in electrical engineering from the Punjab Engineering College, India, in 1998.