Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1997 Proceedings

Americas Conference on Information Systems (AMCIS)

8-15-1997

Information Acquisition for Discrete Resource Allocation: A Comparative Perspective

Raghav Rao H.
State University of New York at Buffalo

Bao-Min Shao
State University of New York at Buffalo

Follow this and additional works at: http://aisel.aisnet.org/amcis1997

Recommended Citation
H., Raghav Rao and Shao, Bao-Min, "Information Acquisition for Discrete Resource Allocation: A Comparative Perspective" (1997). *AMCIS 1997 Proceedings*. 154.
http://aisel.aisnet.org/amcis1997/154

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1997 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Information Acquisition for Discrete Resource Allocation: A Comparative Perspective

H. Raghav Rao and **Bao-Min Shao**
Department of Management Science and Systems
State University of New York at Buffalo
Buffalo, NY 14260

In this paper, two schemes are proposed to facilitate the process of information acquisition for the decision maker in a discrete resource allocation problem (RAP). The RAP, often encountered in artificial intelligence (AI), economics, and operations research, requires the decision maker to recognize the utility functions of agents before the final allocation of resources is made. The acquisition of information on agents' utility functions is achieved by the decision maker through a sequential process that asks the agents about their preference profiles. These two schemes are demonstrated to be effective and compared to show one scheme is theoretically and practically better than the other.

1. Introduction

The discrete resource allocation problem (RAP) examines the problem of how a decision maker can allocate indivisible scarce resources among agents so as to achieve the maximum utility across the agents. A prerequisite for the RAP is that the decision maker should find out about the utility functions of agents on possible resource bundles. In the context of AI, agents can be viewed as any intelligent systems (Russel and Wefald, 1991). Two assumptions can be made here about the agents' utility functions: (I) More units of a resource are preferred to less units of the same kind on the *ceteris paribus* condition that all others remain the same. (II) All agents are rational, i.e., the axioms of reflexivity, transitivity, and completeness are satisfied (Moore, et al., 1994, 1996; Rao, 1991, 1994). Some *a priori* information can be inferred from these two assumptions, but information on those bundles not *a priori* determined still has to be gathered for the decision maker to fully comprehend an agent's utility function.

A resource space can be used to illustrate the idea. Suppose there are two types of resources A and B, and each type has three units. Figure 1 shows the resource space for this scenario. The arrow on the line represents a preference relation on the two associated bundles. The sixteen resource bundles (nodes) form a partial order relation among themselves. The requirement that the decision maker find out about the agent's utility function amounts to converting this partially ordered graph into a totally ordered list.

The acquisition of information by the decision maker is achieved by asking the agent's preference over two resource bundles between which there is no *a priori* information. Rao (1991) has proposed an optimal algorithm to infer the agent's utility function when there are m units of resource A and 1 unit of resource B. Following his work, we propose two schemes to infer the agents' utility functions when there m units of resource A and n units of resource B. Section 2 presents the two schemes. Section 3 gives a comparison between the two schemes, and Section 4 concludes this paper.

2. Schemes Presentation

Scheme 1 is based on the idea of *merging*. From the top of the resource space, we merge every two rows of size m to form a sorted list of size $2m$. Then we proceed to merge every two of these lists of size $2m$ to obtain a list of size $4m$. The process continues until we reach a final sorted list of size mn where the mn resource bundles are sorted, and the decision maker is able to recognize the agent's utility function.

During the two-way merge process, any *a priori* information can be utilized to reduce the number of comparisons if possible. For example, in Figure 1, (3,2) is known to be preferred to (3,1). During the second phase of merging two lists of size 8 generated in the first phase, it is not necessary to compare (3,1) with those bundles already known to be preferred to (3,2) in (3,2)'s list. We can just move those bundles as well as (3,2) to the final list, and directly compare (3,1) with the bundle less preferred to (3,2) in (3,2)'s list.

Scheme 2 is based on the idea of *ranking*. Starting with the bottom row, from left to right, we determine the rank for each bundle by locating its positions on those columns left to it, from right to left. The corresponding positions on the columns will extend from bottom right to top left.

B

(0,3)	(1,3)	(2,3)	(3,3)
(0,2)	(1,2)	(2,2)	(3,2)
(0,1)	(1,1)	(2,1)	(3,1)
(0,0)	(1,0)	(2,0)	(3,0)

A

Figure 1. A resource space

B

(0,3)	(1,3)	(2,3)	(3,3)
(0,2)	(1,2)	(2,2)	(3,2)
(0,1)	(1,1)	(2,1)	(3,1)
(0,0)	(1,0)	(2,0)	(3,0)

A

Figure 2. Scheme 2 ranks bundle (2,1).

Because the resource space is already ordered in rows and columns, to position a bundle on a column left to it can be greatly facilitated by comparing only those bundles on a column which are *so far* unrelated to the positioned bundle using binary search. For example, in Figure 2, the bundles (0,0), (1,0), (2,0), (3,0), (0,1), and (1,1) are already ranked 1, 3, 6, 8, 2, and 4, respectively, and Scheme 2 now wants to determine the rank for (2,1). It is easy to see that there is no need to position (2,1) on columns 1 and 0 because its positions are already determined (encompassed) by the positions of (2,0) and (3,0) on those columns, and the rank for (2,1) is 7, without involving any comparisons.

Both schemes are effective since they can yield the correct total ordering among all the resource bundles. The remaining question is which scheme is better in terms of their performance. The following section addresses this issue.

3. Comparison

For Scheme 1, the worst case happens when each merge of two lists needs the greatest number of comparisons to obtain the resulting list. That implies Scheme 1 needs at most the number of comparisons equal to $n/2(2m - 1) + n/2^2(2^2m - 3) + + n/2^{\lg n} (2^{\lg n}m - (2^{\lg n} - 1)) = n [(m - 1/2) + (m - 3/2^2) + + (m - (2^{\lg n}-1)/2^{\lg n})] = O(mn \lg n)$. Therefore, the running time for Scheme 1 is $O(mn \lg n)$ for m n.

For Scheme 2, the worst case arises when the ranks of the previously determined bundles provide no help for the subsequent bundles. And that happens when all the bundles on a row position themselves below the left-most bundle of the upper row. In this case, Scheme 2 needs $[m(m+1)/2] \lg n + [m(m+1)/2] \lg(n-1) + + [m(m+1)/2] = [m(m+1)/2] \lg(n!)$ $[m(m+1)/2] \lg(n^n) = [m(m+1)/2]\ n \lg n = O(m^2\ n \lg n)$ comparisons from rows 0 to n.

Therefore, the running time for Scheme 2 is $O(m^2 n \lg n)$ for n m (note that m and n are interchangeable if we rotate the resource space). The time complexity analysis states that Scheme 1 is better than Scheme 2 asymptotically.

An experiment is conducted for the case of 2 units of resource A and 2 units of resource B to have an idea of each scheme's performance. There are 197 possible utility functions in this particular case. The results are summarized in Tables 1 and 2. As shown, Scheme 1 needs a little less total number of comparisons than Scheme 2.

4. Conclusion

The issue of information acquisition for the decision maker in a discrete resource allocation problem has been examined in this paper. Two effective schemes have been proposed to facilitate the inquisitive process of the decision maker by asking agents about their preference between two resource bundles which are not *a priori* determined.

Merging-based Scheme 1 is practically and asymptotically better than ranking-based Scheme 2. However, there are still a number of cases where Scheme 2 outperforms Scheme 1. An area of future research is to investigate under what characteristics Scheme 2 will more quickly determine the agents' utility functions.

# of comparisons	Scheme 1 occurrence	Scheme 2 occurrence
4	48	44
5	67	69
6	58	62
7	24	22

Total comp.	1043	1047
Mean	5.29	5.31
Variance	0.939	0.886

Table 1. Experiment summary for (2,2)

Comp. difference of Scheme 1 and 2	Occurrence
3	2
2	16
1	50
0	69
-1	32
-2	24
-3	4

Table 2. Difference of comparisons

References

Moore, J. C., H. R. Rao., and A. B. Whinston, "Multi-agent resource allocation: An incomplete information perspective," *IEEE Trans. Systems, Man, and Cybernetics*, 24 (1994), 1208 - 1219.

Moore, J. C., H. R. Rao., and A. B. Whinston, "Information processing for a finite resource allocation mechanism," *Economic Theory*, 8 (1996), 267 - 290.

Rao, H. R. "An information and preference theory approach to a discrete resource allocation problem," *Automatica*, 27 (1991), 837 - 843.

Rao, H. R. "Two schemes for information acquisition: An entropic assessment," *Automatica*, 30 (1994), 805 - 816.

Russell, S. and E. Wefald, *Do the Right Thing: Studies in Limited Rationality*. The MIT Press, MA (1991).