Instructional Manual for Multilateral Bargaining in Public Decision Making

Glenn W. Harrison and Leo K. Simon

GATT Research Paper 90-GATT 28

Center for Agricultural and Rural Development
Iowa State University
Ames, Iowa 50011

Glenn W. Harrison is with the department of economics, College of Business Administration, University of South Carolina, Columbia. Leo K. Simon is with the department of agricultural and resource economics at the University of California, Berkeley.

This material is based upon work supported by the Cooperative State Research Service, U.S. Department of Agriculture, under Agreement No. 89-38812-4480. Any opinion, findings, conclusions, or recommendations expressed in this publication are those of the author and do not necessary reflect the view of the U.S. Department of Agriculture.

1. INTRODUCTION

This paper describes a laboratory experimental design that has been developed to assess a new political economy institution for possible use in trade negotiations. The objective of this paper is to explain the implementation of this experimental design in the context of a "horizontal" model of a political economy. This environment is much more complex, and in a sense more "realistic", than the simpler environments considered in our [1990] companion paper.

In Section 2 we briefly describe the general Multilateral Bargaining institution once again, so as to make the present discussion self-contained. In Section 3 we describe the present focus on a "horizontal model". In Section 4 we run through in some detail the numerical solution to one parameterized version of that model, to familiarize the reader with the structure and workings of the model. Finally, in section 5 we detail the use of computer software that has been developed to implement the laboratory experiments with the Multilateral Bargaining institution proposed here.

2. THE MULTILATERAL BARGAINING INSTITUTION

The MB institution can be characterized by a model of noncooperative multilateral bargaining with a central player. The model has n+1 players, called the player set. The zero'th player is distinguished from the others and is called the central player. Players 1 through n are peripheral players.

The players participate in a sequential, multilateral bargaining game, similar in spirit to Rubinstein's classic [1982] bilateral game. Their objective in bargaining is to form a coalition, which is just a subset of the player set, and to choose an m-dimensional vector from a set of feasible vectors, called the choice set and assumed to be compact. The choice set may be different for different coalitions.

The central player is distinguished from the others in that she must be included in every coalition. Each player has a utility function defined on the choice set. We assume that utility functions are continuous and strictly quasi-concave.

Problems of this kind are typically formulated as cooperative games. Cooperative game theorists specify some solution concept that satisfies certain appealing properties and then study the set of choices that satisfy the given criterion. Perhaps the most familiar cooperative solution concept is the Core. In the context of the MB institution, a vector x is in the Core if it is feasible for some coalition and if, for every coalition C, there is no feasible vector that is weakly preferred to x by each member of C and strictly preferred by one member.

Noncooperative bargaining theory differs from cooperative game theory in that it attempts to model the actual process of negotiation, rather than just the outcome of the negotiation. A noncooperative model of multilateral bargaining includes an extensive form, which stipulates a particular set of negotiating rules that players must follow.

A natural research program, referred to as the "Nash Program" after Nash [1953], is to study the cooperative and noncooperative versions of a game in conjunction with each other. First one studies a particular cooperative solution concept, then one asks whether the equilibria (usually the subgame perfect equilibria) of some noncooperative model implement the cooperative solutions. Following this approach, we study the relationship between the Core of various bargaining games and the subgame perfect equilibria of our noncooperative version of these games.

The game has a finite number of periods T, each of which is divided into three sub-periods. In the **first** sub-period a player is chosen by Nature to be the proposer. Nature makes it's choice according to a probability distribution over the player set that is prespecified as part of the description of the game. In the **second** sub-period the proposer announces a coalition, of which he must be a member, and a vector that is feasible for that coalition. In the **third** sub-period the remaining members of the proposed coalition each choose whether to accept or reject the proposed vector. If all accept, the game ends. If not, the next period begins and a new proposer is selected. If agreement is not reached by the T'th period then players receive a predetermined disagreement payoff.[1]

A **strategy** for player i specifies the vector that he will announce in each period if selected to be the proposer, as well as a set of vectors that i will accept in each period if he is a member of a coalition announced by some other proposer. A strategy profile is a list of strategies, one for each player. Each strategy profile defines an outcome for the game, which is just a function assigning to each element of the choice set the probability that the game will end with an agreement to select this vector. Note that only a finite number of these probabilities will be positive. Moreover, these positive probabilities need not sum to unity, since the players may never reach an agreement.[2]

A **subgame perfect equilibrium** for a game is a strategy profile with the property that at every sub-period of the game each player's choice is optimal given the strategies specified by the other players. Every T-period game has a subgame perfect equilibrium. Moreover, this equilibrium is generically unique. A striking feature of the model is that there are equilibria in which players fail to agree until the final rounds of bargaining. An

[1] The game is well specified whether or not there is a central player. However, the presence of the central player guarantees that the model has a solution. Nonetheless, it can sometimes be instructive to compare our model to the corresponding one in which the central player is excluded (see Harrison and Simon [1990; section 2.3]).

[2] We are just describing the strategy space here. Equilibrium outcomes, to be defined momentarily, will not admit disagreements.

equilibrium outcome is the outcome defined by a subgame perfect equilibrium. Note that since agents may fail to agree at the beginning of the game, the equilibrium outcome need not coincide with the distribution over first period proposals.

Our theoretical analysis concerns the equilibrium outcomes of games with an arbitrarily large number of periods. Accordingly, our bargaining model is defined as a sequence of T-period bargaining games, with T growing to infinity. A solution to our bargaining model is a limit of the equilibrium outcomes of the T-period games.

The first major analytical result for our model is that a solution exists. That is, the outcomes for the T-period games always converge as T grows large. It is here that the central player has a crucial role: when there is no player that is a member of every coalition, T-period outcomes will not in general converge.

A second major result is that, generically, this solution is deterministic. More precisely, there is generically a unique vector x with the property that for every epsilon there exists a T sufficiently large that the agreed upon vector in any game with more than T periods is within epsilon of x with probability one. When such a vector x exists we will refer to it as the solution vector.

Our last major result is that the solution vector is always in the Core of the corresponding cooperative game.

3. THE HORIZONTAL MODEL

In this section we develop the horizontal model. We assume that there are five members of the alliance. The members will be indexed by the subscript i. The benevolent wing of the government will be denoted by the subscript b, and the political wing by subscript p.

3.1 Endogenous Variables.

There are nine variables that will be determined endogenously as part of the solution to the multilateral bargaining problem. The first is the non-negative fraction π of the policy bounty that will be allocated to the inferior project. The second and third, denoted by $y = (y_1, y_2)$, refer to the location of the inferior project in two-dimensional real space. The remaining five variables, $s = (s_1, \cdots, s_5)$, specify the shares of the policy burden borne by the five members of the alliance.

3.2 The Main Exogenous Variables

The following variables will be specified exogenously. The **rate of return** on the superior investment project is denoted by r. That is, if one unit of the policy bounty is invested in this project, the "general public" will receive r dollars of revenue. The return on the inferior project depends on two system parameters, as well as the locations of the alliance members and of the investment.

The system parameters are a **sectoral inefficiency factor**, θ, taking a value between zero and one, and a positive **spatial inefficiency factor** β_1. Member i's location is denoted by α_i, a point in two-dimensional real space. There is also a **normalization constant**, β_0, which is determined by the other parameters in a manner described below.

If alliance member i is located at α_i and one unit of the policy bounty is invested in the inferior project located at y, then i will receive a net revenue of $\theta(\beta_0 - \beta_1|y - \alpha_i|^2)$ dollars, where $|v - w|$ denotes the Euclidean distance between v and w, and β_0 will be defined below. That is, the inferior investments productivity for a particular member declines quadratically in the distance between the member and the project. The coefficient β_1 determines the sensitivity of productivity to the location of the project.

Observe that **aggregate** net revenue is maximized when the project is located at the mean of the alliance members' locations. The normalization constant β_0 is chosen so that if there were no spatial inefficiencies[3], and no sectoral inefficiencies[4], then the two investments would be equally productive. Our assumption that $\theta < 1$ guarantees that the inferior project is always less efficient than the superior project.

A central aspect of our scenario is that each member of the alliance prefers π to be as large as possible. To ensure that this

[3] That is, y equals the mean of the α_i's.

[4] That is, $\theta = 1$.

is the case we must restrict attention to the subset of the parameter space in which, in equilibrium, each alliance member earns a positive net revenue from the investment. The requirement for positive revenue is that for each i, and each possible location y of the inferior project, $\beta_0 - \beta_1 | y - \alpha_i |^2$. Since β_0 is determined by other parameters and y is determined endogenously, this restriction is rather complicated. It is easy to verify, however, that no player will ever propose a location for y that lies outside the convex hull of the α_i's.

3.3 Payoff Functions

Prior to the policy reform, **alliance member** i has an initial income of γ units. This amount is reduced by i's share of the policy burden s_i, and increased by i's net revenue from the inferior investment project. This revenue is proportional to the fraction π of the policy bounty that is invested in the inferior project. Finally, player i is risk averse, with a coefficient of relative risk aversion of $\rho_i \in [0,1]$.

The **benevolent wing** of the government is concerned only with maximizing aggregate net revenue. It pays no attention to distributional issues. Its payoff, U_b, is determined by the sum of the revenues earned from the two investment projects. For given π and s, player b's payoff is maximized when y is set to the mean of the locations. Note that by the definition of β_0, u_b is strictly increasing in π whenever θ is less than one.

The objective of the **political wing** of the government is to maximize its political support. This support accumulates or dissipates depending on whether the members of the alliance are satisfied or dissatisfied with the outcome of the negotiation process. Specifically, we assume that each member declares in advance an **influence schedule** which assigns a reward or punishment to each policy vector.

In the present model member i rewards or punishes the government according to whether its payoff from the package exceeds or falls short of some **benchmark** payoff level. The benchmark may be interpreted as the utility member i assigns to the status quo. Thus member i rewards the government if it prefers the selected package to the status quo, otherwise the government is punished. For concreteness, imagine that the currency of rewards and punishments is campaign contributions. Each member declares some base contribution level, and increases this level as a reward, or decreases it as a punishment. The steeper a member's schedule, the more sensitive the political wing will be to the preferences of this member. The slope of this schedule, denoted Ψ_i, measures how influential player i will be. For this reason, we refer to Ψ_i as member i's **influence coefficient**.

Summarizing, the payoff function for the political wing is just the sum of the influence schedules of the various members of the alliance.

4. A NUMERICAL EXAMPLE

In this section we consider in considerable detail an explicit example that has been solved numerically. For simplicity, the political wing of the government (Congress) has been excluded from the negotiations and we have declared that burden shares are nonnegotiable. As a byproduct of this discussion, the reader will be introduced to our computer algorithm for solving the model. Understanding the algorithm will help the reader understand the economic logic of the model.

We will refer frequently the computer output which is displayed as Table 1. The first section of the output lists the parameters of the bargaining problem. There are ten admissible coalitions (numbered from one to ten) and six players (numbered from one to six). Each line beginning "Members of coalition number..." is followed by six columns, specifying which players are included in this coalition. For example, coalition #1 consists of players #2, #3, #4 and #6.

The next lines indicate the locations and utility parameters of the privileged sector members. The locations are distributed symmetrically along the horizontal axis. To calculate player i's access probability, divide his access weight by the sum of the access weights. The utility parameters are identical for all members of the alliance. Note that player #6 is chosen to the proposer with probability one-half. The remaining players are chosen with equal probability.

The remainder of the output summarizes the outcome of negotiations in each round of bargaining. Consider, for example, the seven rows of numbers below the statement "Round #50". The first six rows contain ten columns. For $1 \leq i \leq 6$, the first column of row i is the coalition selected by player i in the last period. The second through fourth columns list the policy vector proposed by i: the second column is the value of π; the third and fourth are, respectively, the horizontal and vertical components of the proposed location of y. Columns five through ten specify the payoff that each player will earn if the corresponding policy vector is accepted. The seventh row lists the expected payoff for each player conditional on reaching the final round of negotiations.

We solve the model by standard dynamic programming techniques, starting from the last round. Our maintained hypothesis is that if no agreement is reached in the final round then each player earns an arbitrarily large negative payoff. Consequently, the optimal response for any proposer in this round is to propose his globally optimal policy vector. In particular, each member of the privileged sector proposes a π-value of unicy and a location for y that coincides with the member's own location. Since any player will accept any proposal rather than incur the low disagreement payoff, the proposer can choose any one of the coalitions of which he is member. When the proposer is indifferent between coalitions, our computer algorithm chooses the one indexed by the larger number.

Now consider the penultimate round of negotiations. A member j of a coalition will accept a policy vector proposed by i in this round if and only if the payoff received by j from the proposal is at least as j's **expected** payoff conditional on reaching the next round. Thus, to determine his optimal proposal player i must solve a separate nonlinear programming problem for each coalition to which he belongs.

In the current example the policy space has only two dimensions, since we have restricted attention to a one-dimensional subspace of location space. Since each coalition has four members, there are three "participation constraints". Generically, then, at most two will be binding. In the computer output a binding constraint is indicated by an asterisk following the corresponding payoff number. In round #49, for example, player #6's participation is the only binding constraint for each of players #1 through #5.

Having solved each of the nonlinear programming problems, player i then picks the coalition that yields him the highest payoff. If the payoff exceeds i's expected payoff conditional on reaching the next round, then i will propose this coalition and the corresponding policy vector. Note that there may be rounds in which member i makes a proposal that is **not** accepted. This can happen for one of two reasons. First, i's best feasible alternative may yield him a lower payoff than his expected payoff conditional on passing to the next round. Second, there may be no proposal available to i that satisfies the necessary participation constraints.

We now consider in detail players' proposals in Round #49. First note that unlike round #50, players #1 through #5 cannot persuade player #6 to accept a π-value of unity. Similarly, #6 cannot persuade any of the other players to accept a π-value of zero. Next, observe that for $1 \leq i \leq 5$, the distance between i's location and the origin is inversely related to the magnitude of the π-value proposed by i. The reason is simple. Each player i is constrained by the need to have player #6 accept his proposal. Player #6's optimal location is the origin. Since players #1 and #5 have a greater conflict than the other players over location, they must compromise more over π in order to secure #6's agreement. Similarly, #2 and #4 must compromise over π more than #3.

Our final comment about this round concerns player #6's choice of coalition. He chooses coalition #10, consisting of players #1, #4 and #5. By symmetry, he could have received the same payoff had he chosen coalition #7, which contains the same members as #10 except that #2 replaces #4. The computer chose coalition #10 because ten is larger than seven. Note that player #6 specifies a positive location to satisfy player #4, whose location is positive. Had the computer chosen coalition #7, player #6 would have proposed a negative location and the signs of all his subsequent offers would be reversed.

The importance of this point will be evident when we come to interpret data from our experiments. It is quite likely that we will encounter multiple predictions in the message-space of the experimental game being associated with unique (expected) payoff-space predictions. Thus it will generally be more transparent to test our predictions in terms of how well subjects approximated their maximum possible payoffs rather than how closely their messages correspond to those precicted by our algorithm.

We now turn to round #48. Our first observation is that the distance between the different players' proposals is smaller than in Round #49. The reason is that, on the one hand, player #6's participation constraint is tighter for players #1 through #5, while on the other hand players #2 and #4 have tighter participation constraints are tighter for player #6.

Next, observe that player #6 switches from switches from coalition ten to coalition seven (i.e., he replaces player #4 with player #2). The explanation for this switch is instructive. Recall that in the following round (#49) the proposals that players plan to make are symmetric about the origin, **except** that player #6 plans to propose a positive location for y. Therefore, player #2's expected payoff conditional on reaching the following period is lower than player #4's and so player #2's agreement is easier to obtain in the current period. Note also that player #6 again proposes a **positive** location for y, even though the new member #2 prefers a negative location. This reflects the fact that player #5's participation constraint in the current period is tighter than player #1's, because player #6 plans to propose a positive location in the following period.

As we proceed backwards from Round #47 to Round #44, the gap between the proposed values of π continues to narrow and the proposed locations for y converge towards the origin. Players #1 through #5 are forced to offer smaller and smaller y-values, because player #6's participation constraint continues to tighten.

For player #6, the reasoning is more subtle: as the proposed locations become less skewed to the right, player #2's participation constraint becomes tighter, while #5's becomes looser, and #6's proposed location for the y's reflects this shift in the relative bargaining strength of #2 and #5. Finally, note that throughout this period of the negotiations, player #6 continues to choose coalition #7 over #10. The reason is that player #2's participation constraint remains lower than #4's, because the distribution of y-locations remains skewed to the right.

In the first round of the game each player specifies a π-value of approximately 0.488, and a location for y of approximately zero. This is therefore the approximate solution to the game.

The calculated solution has several intuitively appealing properties. The game is almost symmetric, with player #6 on the one hand and the remaining players on the other. A slight asymmetry arises because players #1 and #5 are not as cohesive as a single agent. If all of the players were as communally oriented as #3, then the outcome would have been exactly symmetric: π would have been equal to 0.5.

The bargaining strength of the alliance was diluted, however, for two reasons. Because they were concerned with the factional components of their objectives, some of the alliance members were less aggressive than player #3 in negotiating for a higher value of

π. A second and related asymmetry in the model is that #6 is an essential player but #3 is not. Player #3 is #6's toughest opponent, but is excluded from #6's coalition except in the earliest rounds of the game. If we modify the example by making player #3 an essential player, then the solution value for π increases to approximately 0.492. This suggests that roughly one-third of the asymmetry[5] arose because the toughest member of the alliance was not essential, while two-thirds arise because of the diffusion of objectives within the alliance.

[5] The solution to the original game provides for 48.8% of the policy bounty to go to the inferior project. This is 1.2% less than the symmetric solution. If player #3 is essential then this share increases by 0.4% (to 49.2%), which is one-third of 1.2%.

5. COMPUTER SOFTWARE

A number of computer programs have been developed to conduct the laboratory experiments. These programs have been designed to facilitate the evaluation of a wide number of alternative treatments in a range of economic environments. Although several features of our experiments have been "hard-wired" into the software, by and large the user interacts with the programs by means of a single ASCII file defining the type of experiment to be conducted. In this section we outline the sequence of programs to be used (section 5.1) and the way in which the user instructs the computer system to run a particular experiment (section 5.2).

5.1 The Computer Programs

There are five programs that are used to conduct and evaluate an experiment: (i) SOLVE, which solves the game assuming rational players, (ii) INIT, which initialize the experiment; (iii) MAIN, which monitors the actual experiment and keeps track of what each individual is doing at all times; (iv) IND, which presents each individual subject with instructions and all messages during the experiment; and (v) OUTPUT, which generates and processes the raw output of the experiment in a (reasonably) transparent manner.

In brief, one can think of MAIN and IND as the two programs that actually conduct the experiment. MAIN is akin to the traffic policeman that makes sure that all of the IND programs obey certain rules when talking to each other using the computer network. It also keeps all of the IND programs up to speed, prompting them for messages when they are not responding. Finally, MAIN keeps the experimenter appraised of the "real-time" progress of the experiment, in case there is some need for experimenter intervention (e.g., if one subject falls asleep, this will become apparent to the experimenter watching MAIN, even if he in turn is asleep ... there are many loud "beeps" to alert the experimenter to possible problems).

IND is the program that each of the subjects interact with. Apart from giving them all of the messages necessary to conduct the experiment, it prompts them for messages to send back. In terms of the "look and feel" of the experiment, IND is the most important from the perspective of the subjects.

SOLVE provides a complete solution to the MB game for the particular parameters being used in the experiment. If there are simulated players then this program must be run prior to the experiment, since it provides the basis for our simulated responses. For most interesting games the amount of time required for a complete solution is quite high. We strongly recommend that this stage of the experimental design be undertaken several days before the actual experiment, at the very least.[6]

INIT simply sets up the experiment before the actual day of the experiment. There is nothing in INIT that cannot be done by

[6] We have also written a special-purpose computer program to solve these dynamic programming problems. This program is considerably faster than the solutions generated by SOLVE, but is not adequately documented to allow readers to understand the logic of our calculations.

MAIN when actually running the experiment, but for some larger experiments with many computer-simulated players there are advantages in having the experiment completely initialized before the day of the experiment. Specifically, the generation of random allocations of players of specified "types" to particular replications of the game can be time-consuming when there are over 30 subjects; this occurs frequently when we have, say, 19 "human" subjects each playing against 5 "simulated" players (resulting in 114 subjects as far as the program is concerned). INIT also checks that the ASCII input file contains all of the information needed to run the experiment.

The final program, OUTPUT, reads the results of the experiment as generated by MAIN and produces a "report" file that is relatively easy to read and interpret. It can also produce analyses of the output as needed.

In terms of the sequence of usage, the experimenter would begin the actual experiment by running MAIN. To do this he would enter the command "MAIN expid" where "expid" is the identifier of the experiment. The program then looks for a file called expid.CNF, the structure of which is described in section below. This file defines the experiment to be run. The program MAIN is typically run on a computer that subjects do not have visual access to, since it may contain some keywords that subjects should not see (so as to ensure control over their motivation).

After all of the files have been initialized by MAIN, it will "beep" several times and alert the experimenter that he may now log in each of the individual subjects. To do this the experimenter simply gives the command "IND" at each of the computers assigned to subjects.

The only important thing that the experimenter must do at this stage is to enter the experiment ID number for each subject. This is typically a sequence of 1, 2, 3, etc., but need not be in particular experimental configurations explained later. We strongly urge the experimenter to initialize this number rather than allowing the individual subject to, since subjects often make mistakes and this can needlessly delay the beginning of the experiment.

One attractive design feature of the experiments is that one can "re-start" any experiment in progress if there is some reason (such as a temporary power failure) to do so. If it is only an IND program that must be re-started, such as if some idiot subject turned off his computer or entered CTL-ALT-DEL, then this can be done without interrupting the other programs. This IND program will just pick up where it left off. If the MAIN program must be re-started then all of the subjects must be logged in again in the original sequence (i.e., wait until the MAIN program says that they may be logged in). The only difference is that the experimenter must tell MAIN to begin in some period other than period 1. Note that the experiment must be re-started in the period in which the system went down: no data from that period will be reliably stored, although all data from the earlier (completed) periods will be. The software is extremely conservative in saving data, doing so at the end of every period. If the system goes down and the experiment must be terminated for that day, then the data will still be saved for all previous periods providing the disk has not been corrupted.

When the experiment is finished the MAIN program will tell the experimenter how much each subject is to be paid. It is wise to write this information down quickly on the record sheet, so that subjects can be paid and dismissed efficiently. As soon as this information has been recorded, the experimenter could run the program OUTPUT to begin the collation and processing of the results. In this way it is typically possible to get some instant feedback on the experimental outcomes, slating one's natural curiosity!

5.2 Configuring an Experiment

The experimenter must initially generate an ASCII file which contains the configuration of the experiment to be conducted. The structure of this file has been carefully chosen so as to minimize the amount of information that needs to be entered, as well as keeping the file as non-cryptic as possible. In this way we hope that the essential experimental environment under study in any particular experiment can be quickly determined by simply inspecting the relevant configuration file.

All data and information may be entered in "free-format", using upper or lower case as seems natural. By "free-format" we mean that there may be one or more spaces between data, to enhance the readability of the file.

There are a number of "delimiters", enclosed in square brackets [], that tell the software that the next lines contain particular pieces of information. It is the job of the experimenter to define all of the necessary delimiters as well as to enter the relevant data in succeeding lines. The delimiters must begin in column 1, and are "reserved words" that should not be used in square brackets that begin in column 1. There are a number of default options for some of the delimiters, so not all of them are required. However, we urge the use of all delimiters simply to be certain that the experiment has been configured as desired.

It is pedagogically easiest to introduce the structure of a CNF file by means of an example. We will discuss the default options later. The following file is called HORIZ.CNF, and implements a small multilateral bargaining experiment:

```
==> HORIZ.CNF ... configuration file for HORIZONTAL model
[nagents]        ' Number of agents, including Government
7

[agents]         ' Names of agents (up to 60 characters).
White_House
Congress
Group_1
Group_2
Group_3
Group_4
Group_5

[npolicies]      ' Number of policy dimensions
8

[policies]       ' Names of policy dimension (up to 60 characters)
Percent Payment
Horizontal Coordinate
Vertical   Coordinate
Share of Group 1
Share of Group 2
Share of Group 3
Share of Group 4
Share of Group 5

[ngroups]        ' Number of experimental groups (or clones)
6

[simulated]      ' Agent or player ID and an asterisk if simulated
White_House      *
Congress         *
Group_1
Group_2
Group_3
Group_4
Group_5          *

[voting power]        ' Agent or player ID and number of votes
White_House      1
Congress         1
Group_1          1
Group_2          1
Group_3          1
Group_4          1
Group_5          1

[access]  ' Agent or player ID and access prob. (to be normalized)
White_House      5
Congress         5
Group_1          1
Group_2          1
Group_3          1
Group_4          1
Group_5          1

[matched  proposals]  ' Proposals from the same agents over reps?
no

[u-default]      ' Agent or player ID and default utility level
White_House        -9999
Congress           -9999
Group_1            -9999
Group_2            -9999
Group_3            -9999
Group_4            -9999
Group_5            -9999

* NOTE: alternatively, user can enter the [p-default] values

[u-squo]         ' Agent or player ID and status quo utility levels
White_House      1
Congress         1
Group_1          1
Group_2          1
Group_3          1
```

```
Group_4           1
Group_5           1

* NOTE: alternatively, user can enter the [p-squo] values

[u-ideal points]    ' Ideal points of Euclidean Utility function
White_House         0       0
Congress            0       0
Group_1             -1.0    0
Group_2             -0.5    0
Group_3             0       0
Group_4             0.5     0
Group_5             1.0     0

[u-intercept]       ' Intercepts of Euclidean Utility function
White_House         0
Congress            0
Group_1             1
Group_2             1
Group_3             1
Group_4             1
Group_5             1

[u-coefficient]     ' Coefficients of Euclidean Utility function
White_House     0
Congress        0
Group_1         0
Group_2         0
Group_3         0
Group_4         0
Group_5         0

[nperiods]      ' Number of periods per game (T)
5

[nrepetitions]  ' Maximal number of times we play the whole game
10

[time]          ' Maximal number of seconds per period
3600

[shuffle]       '  Shuffle players from  game to game?
yes

[path]          '  Path for all messages (this is system-specific)


[solver]        ' Call to GAMS solver
gams

[government]    ' Indicate whether or not there is a Government
no
```

```
[horizontal]     ' Is this is the Horizontal model?
yes

[sectoral inefficiency factor]
0.999

[spatial inefficiency factor]
1.0

[social return]
1.0

[risk aversion coefficient]
0.5

[influence coefficient]
White_House     0
Congress        0
Group_1         1
Group_2         1
Group_3         1
Group_4         1
Group_5         1
```

Each of the options is now discussed in the sequence presented above.

The first line, defining the name of the file, is not required but is a useful convention to follow so that readers know from a hardcopy printout which file they are reading (in this case, HORIZ.CNF).

The number of agents in the experiment is defined with the **[nagents]** delimiter. The line after the delimiter should simply have a number corresponding to the number of agents, including the "center" or "Government" agents. In this case we have five private agents and two Government agents, hence 7 agents in all.

The **[agents]** delimiter allows the entry of names for each of the agents, one name per line. There should be as many names here as there are agents. In the horizontal model we always identify the "benevolent" arm of government first and the "political" arm of government second. The private agents then follow, in any order. The name of any agent may be up to 8 characters, and cannot have spaces and any other funny ASCII character.

The **[npolicies]** delimiter defines the number of policy dimensions that will be bargained over. The following delimiter, **[policies]**, allows the experimenter to provide names for each of the policies. Again, up to 8 characters are allowed. It is recommended that short policy names such as "tax" and "subsidy" be employed to improve readability for subjects (these names will be used in the screen displays given to subjects).

The **[nplayers]** delimiter defines the number of players in the experiment. This is quite distinct from the number of agents per game, since there will typically be several games played simultaneously in any given experiment. Thus, if we have 20 computer terminals and six agents per game, we will typically play three games each with six agents (one terminal is to be used for the monitor program, MAIN). It is easiest to think of [nplayers] as defining the number of experimental subjects in the session. Note that the number of players must be some integer-multiple of the number of agents defined by the [nagents] delimiter.

Each player, or experimental subject, will have an identification number defined by the **[players]** delimiter. These ID's need not actually be numbers, but this is by far the best way to identify subjects for our purposes.

The **[simulated]** delimiter signals that one or more agents or players will be computer-simulated in the experiment. Following this delimiter is a list of either the agent names or the player names, and an asterisk if the agent/player is to be computer-simulated. The reason for the option here of giving agent or player names is that one or other may be more convenient for different experimental purposes. It is not possible to list some agent names and some player names: one or the other is required. All names must be listed, whether or not the agent/player is to be simulated. The names must also be listed in the same order as defined in the [agents] or [players] fields. If there is no asterisk beside an agent/player name, then that agent/player will be a "human" subject. In the above example, note that the two government agents and the final private agent are being simulated.

The **[voting power]** delimiter signals the number of votes that each agent or player has in the game. Again, it is possible to list either agent names or player names depending on whichever is more convenient. In the above example each of the private agents have one vote.

The **[access]** delimiter signals information on the access probability that each agent/player has. This refers to the probability that the agent will be called upon to make a proposal in any period. In the above example each private agent has equal probability. In this case each of the government agents get to make proposals with five times the frequency of the private agents. The program normalizes all of these probabilities so that they sum to unity: it is often easier simply to enter relative integer weights as is done here.

The **[matched proposals]** delimiter, which admits of a "yes" or "no" option", controls the way in which proposers are selected across different games being played simultaneously. If the "yes" option is invoked then the same agent will be chosen in all of the games being played in the same period. Note that this option does not mean that agents are not randomly chosen in each period (according to the access probabilities), but it does add some experimental control over the independent replications being conducted in any given session.

The **[u-default]** array defines the default utility level for each agent or player. An alternative way to input essentially the same information is to define the default policy values using the **[p-default]** delimiter: the program then computes the implied default utility values using these policy values. If the [p-default] option is used then one simply lists the policy names and their default values. In the above example each player's default utility is set at some arbitrarily large negative number.

The **[u-squo]** delimiter defines the status quo utility levels, in the same manner as the [u-default] field. Again, it is possible to enter **[p-squo]** values defining the status quo level of each of the policies.

Three delimiters are used to specify the Euclidean utility function of each agent: **[u-ideal point]**, **[u-intercept]**, and

[u-coefficient]. The interpretation of these coefficients is documented in the GAMS file generated by the SOLVE program.

The **[nperiods]** delimiter defines the number of periods per game. Each game is played with the same players being assigned to given agent types: thus the same four human subjects would be playing against each other for each game, and then would be "shuffled" if this is requested (see below). In the above example there are 5 periods per game.

The **[nrepetitions]** delimiter defines the maximum number of times that the whole game is played. In the above example, we ask for 10 repetitions. Note that player assignments to agent type will be shuffled from repetition to repetition, not from period to period "within" each game as defined here.

The **[time]** option allows the initialization of a time limit, in seconds, on the length of any period. This option causes the programs to start warning subjects that the time limit is approaching. If the time limit is reached without any proposal or response then the next period is immediately implemented. The maximal number of seconds that may be initialized is 30000, which is just over 8 hours and well beyond the maximal bladder size of any known experimental subject! In effect, then, one can turn the time limit "off" by setting this value arbitrarily high. In this case it is important for the experimenter to be monitoring the MAIN program to see if one or more subjects is not responding (this sometimes happen if subjects do not realize that they are being prompted for an input). In the above example we configure the experiment for 1 hour periods (i.e., 3600 seconds), in effect allowing an unlimited time for subjects to respond.

The **[shuffle]** option controls whether or not players are to be shuffled from repetition to repetition. This is an important option to minimize (practically speaking, to eliminate) "reputation effects" from the same players playing in the same games. Note again that we do want the same players to participate in each game, which may well last a number of periods, but we do not necessarily want the same players to meet each other in game after game (i.e., in all of the repetitions of each game). This option, which admits of "yes" or "no" values, will be crucial to our experimental assessment of reputation effects.

The delimiter **[path]** defines the system-specific path in DOS for all file communications. This path is a DOS sub-directory that all subjects, or at least their computer terminals, can access. It is system-specific to particular microlab configurations, but should be easy to ascertain from a system operator and should be constant from experiment to experiment (in the same lab, of course). In the present example we leave this blank, implying that the default DOS sub-directory is used.

The delimiter **[gams]** defines the call to the GAMS software package on the user's system. This package is used in the SOLVE program to set up a series of non-linear programming problems. This program generates a report file, with the suffix ".REP", that is used by the MAIN and IND programs when implementing computer-simulated strategies. We have also developed a faster program to undertake the same calculations as SOLVE but without having to access GAMS; this program is not documented in this report, but the results have been extensively checked with those generated by the SOLVE program, which is documented.

The **[horizontal]** delimiter tells the software whether or not this is the horizontal model. If the model is not, then enter a "no" here and ignore the remaining delimiters in this example (indeed, the default is to assume that this is not the horizontal model, so this delimiter is itself redundant if one is not running the horizontal model). If the horizontal model is invoked, as here, then the next five delimiters should be entered since they define key parameters in the model. Each of these is self-explanatory from the exposition in sections 3 and 4 above.

All of these options may be entered in any order: the computer program reads in the basic "dimensioning" delimiters first and then passes through the CNF file a second time looking for the other arrays. The INIT program is designed to warn you if any delimiter is not read in correctly by the program -- it will inform you if any array is missing or if "default" values have been assigned.

REFERENCES

Harrison, Glenn W., and Simon, Leo K., "Experimental Assessment of Multilateral Bargaining in a Political Economy with an Autonomous Center", **Unpublished Manuscript**, Department of Economics, University of South Carolina, November 1990.

Nash, John F., "Two-Person Cooperative Games", **Econometrica**, V.21, 1953, pp.128-140.

Rubinstein, Ariel, "Perfect Equilibrium in a Bargaining Model", **Econometrica**, v.50, January 1982, pp.97-109.

TABLE 1

A Numerical Example

```
Number of rounds:                                  50
Number of admissible coalitions:                   10
Number of peripheral players:                       5
Number of government players:                       1

Return on superior project (r):                    20.000
Sectoral inefficiency factor (theta):               0.500
Spatial normalization factor (beta(0)):             4.500
Spatial  inefficiency factor (beta(1)):             1.000

Members of coalition number  1:                      OUT      IN       IN        IN       OUT      IN
Members of coalition number  2:                      OUT      IN       IN        OUT      IN       IN
Members of coalition number  3:                      OUT      IN       OUT       IN       IN       IN
Members of coalition number  4:                      OUT      OUT      IN        IN       IN       IN
Members of coalition number  5:                      IN       IN       IN        OUT      OUT      IN
Members of coalition number  6:                      IN       IN       OUT       IN       OUT      IN
Members of coalition number  7:                      IN       IN       OUT       OUT      IN       IN
Members of coalition number  8:                      IN       OUT      IN        IN       OUT      IN
Members of coalition number  9:                      IN       OUT      IN        OUT      IN       IN
Members of coalition number 10:                      IN       OUT      OUT       IN       IN       IN

Location of player number  1: (alpha)             -1.000    0.000
Location of player number  2: (alpha)             -0.500    0.000
Location of player number  3: (alpha)              0.000    0.000
Location of player number  4: (alpha)              0.500    0.000
Location of player number  5: (alpha)              1.000    0.000

Vector of access weights (w):                      1.000    1.000    1.000       1.000    1.000    5.000
Vector of risk aversion coefficients (rho):        0.500    0.500    0.500       0.500    0.500    0.500
Vector of influence coefficients (psi)            0.000    1.000    1.000       1.000    1.000    1.000    0.000
Constant in payoff functions (gamma):             20.000

Convergence tolerance level:                       0.000000010


   Round #50
#10   1.000000 -1.000000  0.000000      4.75395   4.74342   4.71169   4.65833        4.58258   4.63681
#7    1.000000 -0.500000  0.000000      4.74342   4.75395   4.74342   4.71169        4.65833   4.67707
#9    1.000000  0.000000  0.000000      4.71169   4.74342   4.75395   4.74342        4.71169   4.69042
#10   1.000000  0.500000  0.000000      4.65833   4.71169   4.74342   4.75395        4.74342   4.67707
#10   1.000000  1.000000  0.000000      4.58258   4.65833   4.71169   4.74342        4.75395   4.63681
#10   0.000100  0.000000  0.000000      4.56072   4.56072   4.56072   4.56072        4.56072   4.89896
Expected payoffs:                       4.62535   4.64144   4.64678   4.64144        4.62535   4.78130


   Round #49
#8    0.536092 -0.500000  0.000000      4.65954   4.66529   4.65954   4.64225        4.61329   4.78130*
#7    0.561581 -0.238963  0.000000      4.65626   4.66857   4.66883   4.65706        4.63314   4.78130*
#9    0.569598  0.000000  0.000000      4.64730   4.66565   4.67175   4.66565        4.64730   4.78130*
#10   0.561581  0.238963  0.000000      4.63314   4.65706   4.66883   4.66857        4.65626   4.78130*
#4    0.536092  0.500000  0.000000      4.61329   4.64225   4.65954   4.66529        4.65954   4.78130*
#10   0.433440  0.036589  0.000000      4.62535*  4.64007   4.64542   4.64144*        4.62810   4.80966
Expected payoffs:                       4.63363   4.64992   4.65556   4.65060        4.63500   4.79548


   Round #48
#5    0.493228 -0.262003  0.000000      4.64547   4.65582   4.65556*  4.64470        4.62316   4.79548*
#5    0.494631 -0.238963  0.000000      4.64497   4.65584   4.65608   4.64567        4.62457   4.79548*
#9    0.501692  0.000000  0.000000      4.63706   4.65327   4.65865   4.65327        4.63706   4.79548*
#4    0.494631  0.238963  0.000000      4.62457   4.64567   4.65608   4.65584        4.64497   4.79548*
#4    0.493228  0.262003  0.000000      4.62316   4.64470   4.65556*  4.65582        4.64547   4.79548*
```

```
#7    0.484755  0.011911  0.000000       4.63401   4.64992*  4.65538   4.65042        4.63500*  4.79901
Expected payoffs:                        4.63453   4.65049   4.65588   4.65074        4.63502   4.79724


   Round #47
#5    0.490119 -0.159440  0.000000       4.64152   4.65397   4.65588*  4.64725        4.62803   4.79724*
#5    0.490119 -0.159441  0.000000       4.64152   4.65397   4.65588*  4.64725        4.62803   4.79724*
#9    0.493234  0.000000  0.000000       4.63579   4.65172   4.65702   4.65172        4.63579   4.79724*
#4    0.490119  0.159441  0.000000       4.62803   4.64725   4.65588*  4.65397        4.64152   4.79724*
#4    0.490119  0.159440  0.000000       4.62803   4.64725   4.65588*  4.65397        4.64152   4.79724*
#7    0.486992  0.004265  0.000000       4.63467   4.65049*  4.65581   4.65067        4.63502*  4.79854
Expected payoffs:                        4.63482   4.65066   4.65596   4.65075        4.63500   4.79789
```

```
    Round #46
#5    0.488867 -0.101020  0.000000      4.63917   4.65283   4.65596*  4.64858       4.63065   4.79789*
#5    0.488867 -0.101020  0.000000      4.63917   4.65283   4.65596*  4.64858       4.63065   4.79789*
#9    0.490114  0.000000  0.000000      4.63532   4.65115   4.65642   4.65115       4.63532   4.79789*
#4    0.488867  0.101020  0.000000      4.63065   4.64858   4.65596*  4.65283       4.63917   4.79789*
#4    0.488867  0.101020  0.000000      4.63065   4.64858   4.65596*  4.65283       4.63917   4.79789*
#7    0.487615  0.001464  0.000000      4.63488   4.65066*  4.65593   4.65072       4.63500*  4.79841
Expected payoffs:                       4.63493   4.65073   4.65599   4.65076       4.63500   4.79815


    Round #45
#5    0.488366 -0.063934  0.000000      4.63766   4.65209   4.65599*  4.64940       4.63227   4.79815*
#5    0.488366 -0.063934  0.000000      4.63766   4.65209   4.65599*  4.64940       4.63227   4.79815*
#9    0.488865  0.000000  0.000000      4.63513   4.65092   4.65617   4.65092       4.63513   4.79815*
#4    0.488366  0.063934  0.000000      4.63227   4.64940   4.65599*  4.65209       4.63766   4.79815*
#4    0.488366  0.063934  0.000000      4.63227   4.64940   4.65599*  4.65209       4.63766   4.79815*
#7    0.487864  0.000466  0.000000      4.63496   4.65073*  4.65598   4.65075       4.63500*  4.79836
Expected payoffs:                       4.63498   4.65075   4.65600   4.65076       4.63500   4.79826


    Round #44
#5    0.488165 -0.040446  0.000000      4.63669   4.65161   4.65600*  4.64991       4.63328   4.79826*
#5    0.488165 -0.040446  0.000000      4.63669   4.65161   4.65600*  4.64991       4.63328   4.79826*
#9    0.488364  0.000000  0.000000      4.63505   4.65083   4.65608   4.65083       4.63505   4.79826*
#4    0.488165  0.040446  0.000000      4.63328   4.64991   4.65600*  4.65161       4.63669   4.79826*
#4    0.488165  0.040446  0.000000      4.63328   4.64991   4.65600*  4.65161       4.63669   4.79826*
#10   0.487964  0.000339  0.000000      4.63498*  4.65075   4.65600   4.65076*       4.63501   4.79834
Expected payoffs:                       4.63499   4.65076   4.65601   4.65077       4.63500   4.79830


    Round #43
#5    0.488084 -0.025584  0.000000      4.63607   4.65130   4.65601*  4.65023       4.63392   4.79830*
#5    0.488084 -0.025584  0.000000      4.63607   4.65130   4.65601*  4.65023       4.63392   4.79830*
#9    0.488164  0.000000  0.000000      4.63502   4.65079   4.65604   4.65079       4.63502   4.79830*
#4    0.488084  0.025584  0.000000      4.63392   4.65023   4.65601*  4.65130       4.63607   4.79830*
#4    0.488084  0.025584  0.000000      4.63392   4.65023   4.65601*  4.65130       4.63607   4.79830*
#7    0.488004  0.000127  0.000000      4.63499   4.65076*  4.65601   4.65077       4.63500*  4.79833
Expected payoffs:                       4.63500   4.65077   4.65601   4.65077       4.63500   4.79832


    Round #42
#5    0.488052 -0.016182  0.000000      4.63568   4.65111   4.65601*  4.65043       4.63432   4.79832*
#5    0.488052 -0.016182  0.000000      4.63568   4.65111   4.65601*  4.65043       4.63432   4.79832*
#9    0.488084  0.000000  0.000000      4.63501   4.65078   4.65602   4.65078       4.63501   4.79832*
#4    0.488052  0.016182  0.000000      4.63432   4.65043   4.65601*  4.65111       4.63568   4.79832*
#4    0.488052  0.016182  0.000000      4.63432   4.65043   4.65601*  4.65111       4.63568   4.79832*
#10   0.488020  0.000081  0.000000      4.63500*  4.65077   4.65601   4.65077*       4.63500   4.79833
Expected payoffs:                       4.63500   4.65077   4.65601   4.65077       4.63500   4.79832
                                                                                     Round #3
#5    0.488031 -0.000593  0.000000      4.63503   4.65078   4.65601*  4.65076       4.63498   4.79833*
#5    0.488031 -0.000595  0.000000      4.63503   4.65078   4.65601*  4.65076       4.63498   4.79833*
#9    0.488031  0.000000  0.000000      4.63500   4.65077   4.65601   4.65077       4.63500*  4.79833*
#4    0.488031  0.000595  0.000000      4.63498   4.65076   4.65601*  4.65078       4.63503   4.79833*
#4    0.488031  0.000593  0.000000      4.63498   4.65076   4.65601*  4.65078       4.63503   4.79833*
#10   0.488031  0.000000  0.000000      4.63500*  4.65077   4.65601   4.65077*       4.63500*  4.79833
Expected payoffs:                       4.63500   4.65077   4.65601   4.65077       4.63500   4.79833


    Round #2
#5    0.488031 -0.000593  0.000000      4.63503   4.65078   4.65601*  4.65076       4.63498   4.79833*
#5    0.488031 -0.000595  0.000000      4.63503   4.65078   4.65601*  4.65076       4.63498   4.79833*
#9    0.488031  0.000000  0.000000      4.63500   4.65077   4.65601   4.65077       4.63500*  4.79833*
#4    0.488031  0.000595  0.000000      4.63498   4.65076   4.65601*  4.65078       4.63503   4.79833*
#4    0.488031  0.000593  0.000000      4.63498   4.65076   4.65601*  4.65078       4.63503   4.79833*
#10   0.488031  0.000000  0.000000      4.63500*  4.65077   4.65601   4.65077*       4.63500*  4.79833
Expected payoffs:                       4.63500   4.65077   4.65601   4.65077       4.63500   4.79833
```

```
   Round #1
#5     0.488031 -0.000593  0.000000      4.63503    4.65078    4.65601*  4.65076         4.63498    4.79833*
#5     0.488031 -0.000595  0.000000      4.63503    4.65078    4.65601*  4.65076         4.63498    4.79833*
#9     0.488031  0.000000  0.000000      4.63500    4.65077    4.65601   4.65077         4.63500*   4.79833*
#4     0.488031  0.000595  0.000000      4.63498    4.65076    4.65601*  4.65078         4.63503    4.79833*
#4     0.488031  0.000593  0.000000      4.63498    4.65076    4.65601*  4.65078         4.63503    4.79833*
#10    0.488031  0.000000  0.000000      4.63500*   4.65077    4.65601   4.65077*         4.63500*  4.79833
Expected payoffs:                        4.63500    4.65077    4.65601   4.65077         4.63500    4.79833
```