Online Appendix: Getting Permission

Peicong Hu* and Joel Sobel†

January 25, 2022

*Peicong Hu: Faculty of Business and Economics, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: kerihu@hku.hk.

†Joel Sobel: Department of Economics, University of California, San Diego, La Jolla, CA 92093, U.S.A. E-mail: jsobel@ucsd.edu.

Appendix: Sequential Protocols

In this appendix we give an example that shows the importance of examining protocols that involve some commitment and repeated consultations. We also describe general sequential consultation procedures formally.

Example

The definition of sequential protocols permits the manager to do three things: vary the order in which he consults experts; return to experts more than once; and commit to ending the consultation process. The following example demonstrates why these three features are important and gives some insight into the general construction.

Example 1. *There are five projects,* $0, 1, \ldots, 4$*. The projects are completely ordered and the manager prefers higher projects to lower ones. Project* 0 *is the status quo. There are two experts. Expert 1's utility satisfies*

$$u_1(1) > u_1(3) > u_1(0) > u_1(2) > u_1(4)$$

and Expert 2's utility satisfies

$$u_2(0) > u_2(2) > u_2(1) > u_2(4) > u_2(3).$$

The unique Nash equilibrium outcome in the simultaneous game is the manager's favorite outcome.

Consider the four possible consultation sequences that consult each expert at most once: consulting exactly one expert, or consulting both in either order.

Sequence	*Outcome*
Expert 1	1
Expert 2	0
Expert 1, then 2	0
Expert 2, then 1	1

The first two lines in the table are straightforward to understand. When the manager consults only one expert, the expert picks her favorite project. If he instead consults Expert 1 and then Expert 2, Expert 2 will approve project 2 if Expert 1 starts with 1; Expert 2 will approve project 4 if Expert 1 starts with 3; Expert 2 will approve project 0 if Expert 1 starts with 0; if Expert 1 starts with 4, Expert 2's action will not influence the project choice; and Expert 2 will not approve a project 3 or 4 if Expert 1 starts with 2. Hence Expert 1 does best if she approves the status quo. Similarly, if the manager asks Expert 2 first, then Expert 1, the final outcome will never be 0 or 2. So Expert 2 supports project 1 and Expert 1 does not support a higher project.

It is straightforward to confirm that returning to experts will not lead to either expert supporting another project. Hence, it appears that sequential consultation need not lead to support for π^*.

We will investigate the implications of giving the manager more control over the nature of consultation. Suppose the manager begins by asking Expert 1 "Will you support Project 4?" If Expert 1 says "yes," then the manager stops and implements his favorite project. If Expert 1 declines (but possibly supports another project), then the manager repeats the question to Expert 2. If Expert 2 declines, then the manager returns to Expert 1 and requests approval for project 3. And so on: the manager consults experts one-by-one, asking for a support. If both experts decline, then the manager returns to the first expert and asks for approval of the next best project.

When the experts play this game, we can work backwards to see that an expert will approve Project 4. Suppose that both experts have rejected Projects 2, 3, and 4. Expert 1 would support Project 1, because that is her favorite. Knowing this, Expert 2 will support Project 2 because she knows that Project 0 is not available (because Expert 1 will support Project 1). But Expert 1 prefers Project 3 to Project 2, so she will support Project 3 when asked. Finally, given that Expert 1 would support Project 3 if asked, Expert 2 will support Project 4.

Sequential Protocols

In each period t, the procedure selects an expert to make a choice. When chosen, an expert specifies a project (which could be the status quo). Next, the manager either consults with another expert or stops. This subsection describes the set of protocols formally and discusses the game defined by a protocol.

Formally, let $H_0 = \varnothing$, $H_t = (I_+ \times X)^t$, where X is the (finite) set of possible projects and $I_+ = \{0\} \cup I$ is the union of "0" and the set of players. Let $H = \bigcup_{t=0}^{T} H_t$ be the set of histories, where T is a positive integer that bounds the total number of consultations. If $h_t = (h_t^1, \ldots, h_t^t) \in H_t$ and $h_{t'} = (h_{t'}^1, \ldots, h_{t'}^{t'}) \in H_{t'}$, then $h_t h_{t'} \in H_{t+t'}$ is the history obtained by the natural concatenation: $h_t h_{t'} = (h^1, \ldots, h^t, h^{t+1}, \ldots, h^{t+t'})$ where

$$h^m = \begin{cases} h_t^m & \text{if } 1 \leqslant m \leqslant t \\ h_{t'}^{m-t} & \text{if } t < m \leqslant t + t'. \end{cases}$$

Definition. *A sequential protocol is a mapping $P : H \to I_+$ such that for all $h, h_t \in H$,*

$$P(h_T) = 0 \textit{ for all } h_T \in H_T, \tag{1}$$

$$P(h_t) = 0 \Longrightarrow P(h_t h) = 0, \tag{2}$$

The manager observes a history, h_t. He then decides whether to stop the process ($P(h_t) = 0$) or to consult Expert i ($P(h_t) = i$). Condition (1) expresses the fact that the protocol must stop after T periods. Condition (2) means that once the decision maker stops the process, he cannot restart it.

A sequential protocol (we shorten this to "protocol") induces a perfect-information game in which the players are the experts. Player i's strategy specifies a project as a function of h_t for each h_t such that $P(h_t) = i$. Given a history of length t, $h_t = (h_t^1, \ldots, h_t^t)$, let $i_t(h_t) = (i_t^1, \ldots, i_t^t)$ be the list of experts consulted and $p_t(h_t) = (p_t^1, \ldots, p_t^t)$ be the list of projects supported, and let $\mu(h_t) = \max\{p_t^1, \cdots, p_t^t\}$. A strategy profile $\mathbf{s} = (s_1, \ldots, s_I)$ determines projects $\bar{p}_t(\mathbf{s}) = (\bar{p}_t^1(\mathbf{s}), \ldots, \bar{p}_t^t(\mathbf{s}))$ and histories $\bar{h}_t(\mathbf{s})$ for $t = 1, \ldots, T$ where $\bar{h}_1(\mathbf{s}) = (P(\varnothing), \bar{p}_1(\mathbf{s})) = (P(\varnothing), s_{P(\varnothing)}(\varnothing))$,

$\bar{p}_2(\mathbf{s}) = (\bar{p}_1(\mathbf{s}), s_{P(\bar{h}_1(\mathbf{s}))}(\bar{h}_1(\mathbf{s})))$, $\bar{h}_2(\mathbf{s}) = (\bar{h}_1(\mathbf{s}), (P(\bar{h}_1(\mathbf{s})), \bar{p}_2^2(\mathbf{s})))$, and, in general, $\bar{p}_k(\mathbf{s}) = (\bar{p}_{k-1}(\mathbf{s}), s_{P(\bar{h}_{k-1}(\mathbf{s}))}(\bar{h}_{k-1}(\mathbf{s})))$, $\bar{h}_k(\mathbf{s}) = (\bar{h}_{k-1}(\mathbf{s}), (P(\bar{h}_{k-1}(\mathbf{s})), \bar{p}_k^k(\mathbf{s})))$.[1] Expert i's payoff as a function of the strategy profile is $\tilde{u}_i(\mathbf{s}) = u_i(\mu(\bar{h}_T(\mathbf{s})))$. We say that a project π is generated by a sequential protocol if the induced game has a strategy profile that survives IDWDS in which π is implemented. Formally, π is generated by a sequential protocol if there exists a strategy profile $\mathbf{s}$ that survives IDWDS such that $\pi = \max\{p_T^1(\mathbf{s}), \ldots, p_T^T(\mathbf{s})\}$. A project π is uniquely generated by a sequential protocol if π is the only project generated by the protocol.

The specification of the game assumes that the manager implements $\mu(\bar{h}_T(\mathbf{s}))$. One can imagine games in which the manager does not do this. If the manager implements $\gamma(h)$ given the history h, then our specification of the game requires setting $\tilde{u}_i(\mathbf{s}) = u_i(\gamma(\bar{h}_T(\mathbf{s})))$ and a project π would be generated by a sequential protocol if there exists a strategy profile that survives IDWDS such that $\pi = \gamma(\bar{h}_T(\mathbf{s}))$.

We next describe the canonical protocol formally.

Assume that the manager has strict preferences over projects. Specifically, suppose that the projects can be ranked $\bar{x} = \pi_K \succ \pi_{K-1} \succ \cdots \succ \pi_1 = \underline{x}$.

Definition. *The canonical sequential protocol (CP) has the properties*

- $T = KI$

  and if $t = mI + r$, *for* $r = 0, \ldots, I-1$ *and* $m = 0, \ldots, K-1$, *then*

- $$P(h_t) = \begin{cases} 0 & \text{if } \mu(h_t) \geq \pi_{K-m} \\ r+1 & \text{otherwise} \end{cases}.$$

In the canonical sequential protocol (CP), the manager first consults Expert 1, who can support any project. If the expert supports a "good enough" project, then the procedure stops and this project is implemented. Otherwise, the manager consults the next expert, returning to Expert 1 if $i = I$ (and Expert I did not support a good enough project). The first time an expert is consulted, only project π_K is

[1] These formula require a specification of s_0 because it is possible that $P(h) = 0$. We set $s_0(h) = \mu(h)$.

"good enough" to stop the procedure. In general, project π_{K-k} is good enough when all experts have been consulted (at least) k times.

CP depends directly on the manager's preferences (that is, the order of projects in the protocol depends on $\succ$). The preferences of the manager do play a role in the simultaneous-move game ($M(\mathbf{x})$ determines payoffs and $M(\mathbf{x})$ depends on $\succ$), but the strategies in the simultaneous-move game do not depend on the manager's preferences. The canonical sequential protocol does not require that the manager have any information about the experts' preferences.