Available online at www.sciencedirect.com

ScienceDirect

Journal of Economic Theory 160 (2015) 175–187

JOURNAL OF Economic Theory

www.elsevier.com/locate/jet

Notes

Dynamically consistent voting rules

Madhav Chandrasekher

Department of Economics, Arizona State University, Main Campus, PO Box 879801, Tempe, AZ 85287-9801, United States

Received 17 January 2014; final version received 18 August 2015; accepted 30 August 2015

Available online 5 September 2015

Abstract

This paper studies preference aggregation in a dynamic choice context. Voters face menus of options in stages and the source of dynamics is that menus possibly get smaller across each stage. We call the family of stage voting rules "dynamic voting rules" and provide an axiomatic characterization, on the domain of single-peaked preferences, of dynamic voting rules that are strategy-proof and satisfy a second property, inspired from choice theory, which we call dynamic consistency.

JEL classification: D03; D60; D70; D71; D81; D90

Keywords: Aggregation of preferences; Dynamic consistency; Strategy-proofness

1. Introduction

This paper conducts an axiomatic analysis of dynamic voting rules. To illustrate the sort of choice problem we have in mind, imagine a firm has an open position and that it fills the position by a vote of the partners. First, the firm solicits applications and forms a candidate pool, on which it votes. When the firm makes an offer to its first choice, the candidate may not accept the job, and candidates may exit the pool (e.g., because they have taken a position at another firm). When a candidate exits the pool the firm faces a smaller pool after the first offer, and subsequently smaller pools after subsequent offers. This is an example of a dynamic aggregate choice problem. Decision nodes are indexed by the currently available pool of candidates and at

E-mail address: madhav.chandrasekher@asu.edu.

http://dx.doi.org/10.1016/j.jet.2015.08.012

each node committee votes are aggregated into an aggregate decision (e.g., making an offer to a candidate). The rule that governs how committee votes are used to form an aggregate decision (e.g., majority rule, super-majority rule, unanimity rule) is called the voting rule and, in theory, each decision node could be governed by a different voting rule.

Two conditions will restrict the collection of voting rules across decision nodes. First, they must induce truthful reporting of preferences at every decision node. This requirement is common in the voting literature and is referred to as "strategy-proofness". Strategy-proofness formalizes the idea that voting rules should be non-manipulable. The second condition is that the voting rules used to reach an aggregate decision at each node must be consistent with one another (e.g., the firm always makes the offer to the best qualified candidate in each round). We refer to this second condition as "dynamic consistency". The concept of dynamic consistency is borrowed from decision theory. To explain the idea, consider a choice problem under uncertainty where the uncertainty resolves in stages and, at each stage, a decision-maker's (interim) preferences adapt to the new information. Dynamic consistency disciplines the collection of interim preferences in the following manner (see, e.g., Siniscalchi, 2011): if one choice is interim preferred to another no matter which way uncertainty resolves, then this preference is maintained on the node immediately prior to when this uncertainty resolves. In our decision problem there is no uncertainty. Nonetheless, we can extrapolate from this a general principle of dynamic consistency: if there is no payoff relevant change in the choice environment between two successive decision nodes, then whenever one choice is preferred to another at a successor node this preference is maintained at its predecessor node.

We use this principle to define dynamic consistency in our context. The time-dependent objects in our setting are the choice pools (menus). Dynamic consistency will specify a condition under which the aggregate choice from a larger menu, say B, is the same as the subsequent choice from some menu A, where A is contained in B. Our condition is a formal generalization of Sen's criterion α. The application Sen had in mind when he formulated his axiom (see Sen, 1969) was the problem of rationality of social choice functions (the problem being non-existence of transitive social preferences, after Arrow). To describe Sen's α, imagine option pools are considered in sequence as choices are removed from an initial set. Sen's criterion states that if the choice from some initial set B is still present in a subsequent (and smaller) option set A, then the same choice is made from A. We generalize this by requiring the aggregate choice to agree across two points in time so long as the original choice is still available *and* each voter's first-best alternative from the larger option set is still present in the smaller option set. This is our definition of dynamic consistency.

The normative justification for dynamic consistency is that it links sequential decision-making to the idea of rational choice, i.e., choices are derived from preference maximization. Importantly, in the single-agent setting dynamic consistency adds "some" rationality to the decision-making process without forcing choices to come from preference maximization. In the choice under uncertainty example, there is a different preference relation at each decision node and the supports of these relations are distinct, so it wouldn't make sense[1] for there to be a single, over-arching preference relation that is maximized at each node. In the sequential setting of Sen's α, requiring choice to be rational (preference maximization) makes sense but it also frustrates Sen's goal of studying problems in which the "context" (the option set itself) influences a decision-maker's preferences. When we shift to the aggregate, however, the question of rationality – equivalently,

1 Excepting uninteresting cases, e.g., all acts (consumption plans) are indifferent or uncertainty is trivial.