Monotone Implementation*

Levent Ülkü[†]
CIE and Department of Economics
ITAM

November 2009

Abstract

We study a class of direct revelation mechanisms which implement outcome functions satisfying a monotonicity condition. Monotone implementation is in dominant strategy equilibrium when values are private and in ex post Nash equilibrium when values are interdependent. The original Vickrey-Clarke-Groves mechanism is not a monotone implementation mechanism although its many extensions to interdependent value models are. The extraction mechanisms of Cremer and McLean (1985) are a special form of monotone implementation mechanisms for finite type spaces.

1 Introduction

Research on implementation under incomplete information largely focuses on efficient social outcomes. The leading formulation of the efficient mechanism design problem is with private values and quasilinear utilities, and

*Work in progress. Current versions are at http://allman.rhon.itam.mx/levent.ulku. Previously circulated under the title: Ex Post Implementation over Lattices.

[†]levent.ulku@itam.mx

the Vickrey-Clarke-Groves (VCG) mechanism induces dominant strategy incentives for truthful reporting of private information. This result critically depends on the private values assumption and efficiency of the social decision. In this paper we show that under the quasilinearity assumption social outcomes satisfying a monotonicity property are implemented in ex post Nash equilibrium under interdependent values (in dominant strategy equilibrium if values are private) by a class of monotone implementation mechanisms.

The VCG mechanism implements an efficient outcome by asking each agent for a payment equal to the cost he imposes on the rest of the agents. For any agent i, this cost is exactly the difference between the joint welfare of the rest of the society if i did not participate in the mechanism and their joint welfare under the efficient outcome with i's participation. Under private values neither of these magnitudes depends on i's private information. As a consequence no agent has an incentive to deviate from truthful reporting regardless of the reporting strategy followed by other agents. Suppose that by misreporting his type an agent can change the collective decision from the efficient outcome to a different outcome. His net utility gain is the resulting change in social welfare which is necessarily nonpositive. Remarkably, this reasoning does not rely on the exact nature of private information and how it affects valuations and marginal valuations for the outcome variable. Equally importantly, nor does it depend on any order structure on the set of outcomes, or an associated monotonicity/comonotonicity condition on valuations.

There are important mechanism design problems, however, where values are interdependent and/or the mechanism designer's objective deviates from efficiency. In such problems the VCG mechanism does not have desirable incentive properties. In this paper we show that if the outcome function satisfies an "individual monotonicity" property, then incentive compatibility may still be attained despite deviations from the private values-efficient mechanism design paradigm. An outcome function is *individually monotone* if it improves according to an agent-specific criterion when that agent's type goes up and all other types remain the same. Individual monotonicity is an appealing feature of social outcomes and it follows from standard single crossing conditions in many mechanism design problems of interest.

Consider a deviation from the goal of efficiency in a private values framework. Suppose that the mechanism designer's objective is to choose outcome q to maximize a weighted sum $\sum_{j=1}^{n} \lambda_j v_j(q, t_j)$ where n is the number of agents, $v_j(q, t_j)$ is agent j's value for q when his type is t_j and $\lambda_j \geq 0$ is a welfare weight. If the welfare weights are not all equal then the resulting outcome function $(t_1, ..., t_n) \mapsto q^{\lambda}(t_1, ..., t_n)$ is not efficient. In the spirit of the VCG mechanism, consider combining q^{λ} with payments that charge each agent the cost he imposes on the rest of the agents. Under this mechanism, any agent who changes $q^{\lambda}(t)$ to q' by misreporting his type achieves a net gain of $\sum_j v_j(q', t_j) - \sum_j v_j(q^{\lambda}(t), t_j)$. Since q^{λ} is not efficient, this difference may be positive and the mechanism is not incentive compatible. Our main result will indicate that if (1) type spaces are intervals or finite and completely ordered, (2) the outcomes are partially ordered by every agent, (3) valuations satisfy monotone differences, and (4) q^{λ} satisfies individual monotonicity, then payments can be constructed which induce truthful reporting in a dominant strategy equilibrium.

A more fundamental change in the model entails the introduction of value interdependence. Suppose that valuations depend on the collective type profile $t = (t_1, ..., t_n)$ and the mechanism designer wishes to maximize social welfare $\sum_{j=1}^{n} v_j(q, t)$ at every t. Now the social cost imposed by i on the rest of the agents critically hinges on how t_i would be inferred and used if i was hypothetically removed from the society. Hence there is a conceptual difficulty in defining payments equal to social costs in interdependent value models.[1] Despite this complication, solutions to the efficient mechanism

[1] A straightforward extension of the VCG mechanism which calculates social cost imposed by i using t_i is in general not incentive compatible. Incentivizing truthful revelation of private information for an agent removed from the society is difficult. McLean and Postlewaite (2006) show that approximate ex post incentive compatibility can neverthe-

design problem under interdependent values have been obtained in various papers (Cremer and McLean [1985], Ausubel [1999], Jehiel and Moldovanu [2001], Bergemann and Valimaki [2002]). These solutions are direct revelation mechanisms commonly referred to as "generalized" VCG mechanisms. Our analysis complements this literature by highlighting the role of the individual monotonicity condition in attaining incentive compatibility in the efficient mechanism design problem with interdependent values.

Let us sketch the results and the procedures commonly used in the aforementioned papers. The "generalized" VCG mechanisms share important properties. First, they are ex post Nash incentive compatible[2] if the efficient outcome rule is monotone. Monotonicity is not a logical consequence of efficiency alone but it follows under special assumptions which we discuss below. Second, if values are private, then a generalized VCG mechanism charges each agent the original VCG payments. The following features of these models are worth special emphasis.

- Type spaces are intervals on the real line, or as in Cremer and McLean [1985], they are finite and completely ordered. This is not a coincidence as Jehiel and Moldovanu [2001] and Jehiel et al. [2006] show that very little can be accomplished in terms of incentive compatibility in interdependent value models with general multidimensional type spaces.

- The set of social decisions is ordered and this has a bearing on the definition of important monotonicity, interagent crossing and monotone differences conditions.[3]

less be obtained in large societies as agents become informationally small and their private information can be inferred by the rest of the agents with accuracy.

[2]Ex post Nash equilibrium is a solution concept for Bayesian games which requires the play of Nash equilibrium at every resolution of uncertainty. A mechanism is ex post Nash incentive compatible if truth telling is an ex post Nash equilibrium in the game it induces. Ex post Nash incentive compatibility is now the standard notion of incentive compatibility in mechanism design with interdependent values.

[3]To juxtapose the original VCG mechanism with its generalizations, we should reem-

Building on these features two important conditions play crucial roles in the formulation of generalized VCG mechanisms: (1) an interagent crossing condition which states that a change in t_i induces a larger change in i's marginal value for the social outcome then it does in that of agent j's, and (2) a monotone differences condition which states that higher types of an agent have higher marginal values for the social outcome. The interagent crossing condition serves the purpose of obtaining an efficient outcome function which satisfies individual monotonicity.[4] The monotone differences property, in conjunction with the individual monotonicity of the efficient outcome function, implies that incentive compatible payments can be constructed. In a sense, then, the generalized VCG mechanisms constitute a generalization of a special subclass of VCG mechanisms which implement efficient *and* monotone social outcomes. Thus they belong to the class of monotone implementation mechanisms which we analyze in this paper.

The framework we have described is common to the analyses of auctions in Cremer and McLean (1985) and Ausubel (1999) and the abstract social choice model in Jehiel and Moldovanu (2001). In the Jehiel-Moldovanu model, valuations are linear in the type vector and the interagent crossing condition takes an especially simple form: the marginal effect of an agent's type on social welfare must be increasing in the social alternative. In related work, Bergemann and Valimaki (2002) show that it is the monotonicity of the efficient outcome that leads to its implementability, without using an interagent crossing condition. Our results will indicate that if monotonicity is satisfied, then efficiency does not play any role in the implementability of a social outcome.

Monotone implementation is commonly used in a host of models which

phasize that neither the definition, nor the incentive compatibility of the VCG mechanism depends on how private information is modeled and how social alternatives are ordered.

[4]The interagent crossing condition also appears in Maskin [1992], Maskin and Dasgupta [2000] and Perry and Reny [2002] indicating that general auction mechanisms also require monotonicity in the outcome variable.

are not primarily centered around the goal of efficiency. All studies of the optimal auction problem make assumptions that guarantee the individual monotonicity of the optimal auction. (See Myerson [1981], Branco [1996] and Levin [1997] among others.) The generalized second price auction used in internet advertising (Edelman et al. [2006]) satisfies our monotonicity condition in the sense that advertiser i gets a better spot if his bid goes up, other bids remaining constant. Bergemann and Morris [2007] present an interdependent value auction which is strictly monotone and approximately efficient. Their auction can be implemented in strict dominant strategy equilibrium by a monotone implementation mechanism.

The paper is structured as follows. In section 2 we analyze monotone implementation in two environments where the VCG mechanism is not incentive compatible. In Section 3 we outline the model and give our main result concerning the implementability of individually monotone outcome functions. In Section 4 we identify sufficient conditions for efficient outcome functions to satisfy individual monotonicity. Section 6 is devoted to an analysis of a class of set allocation problems where a different set of sufficient conditions apply. Section 7 concludes. The proofs are collected in the appendix.

2 Examples

In this section we illustrate monotone implementation mechanisms in two examples. First, we analyze a private value environment where the mechanism designer solves a weighted social welfare maximization problem. The second example is a standard interdependent value auction.

Example 1 (Weighted welfare maximization under private values) Consider the following stylized environment. The agents are indexed by $i = 1, ..., n$ and C is a set of social outcomes. Each agent i is equipped with a type space T_i and a valuation function $v_i : C \times T_i \to \Re$. If agent i's type is t_i, he makes a payment x_i and the outcome is c, then i's utility is $v_i(c, t_i) - x_i$. The set of collective type vectors is $T = T_1 \times \cdots \times T_n$. An *outcome function* is a map $q : T \to C$. For now, we will say that q is *implementable* (in dominant strategy equilibrium) if there exists $x : T \to \Re^n$ such that for every $i, t = (t_i, t_{-i})$ and $t'_i \neq t_i$, we have $v_i(q(t), t_i) - x_i(t) \geq v_i(q(t'_i, t_{-i}), t_i) - x_i(t'_i, t_{-i})$. (We will strengthen the notion of implementability to require ex post individual rationality and to rule out ex post budget deficits in the next section.) Let $\lambda_1, ..., \lambda_n$ be nonnegative scalars not all equal to zero and suppose that $q^\lambda(t)$ solves $\max_c \sum_i \lambda_i v(c, t_i)$ for every $t \in T$. We will investigate the implementability of q^λ.

Consider the "VCG payments"

$$x_i^{VCG}(t) = \max_c \sum_{j \neq i} v_j(c, t_j) - \sum_{j \neq i} v_j(q^\lambda(t), t_j)$$

for every i and t. Let $U_i(t'_i, t'_{-i}|t_i)$ be the ex post payoff to i if the type profile is $t = (t_i, t_{-i})$, the rest of the agents report t'_{-i} and he reports t'_i. Note that under private values this number does not depend on t_{-i}. Also define $V(c, t) = \sum_{j=1}^n v_j(c, t_j)$. We have

$$\begin{aligned} & U_i(t'_i, t'_{-i}|t_i) - U_i(t_i, t'_{-i}|t_i) \\ & = [v_i(q^\lambda(t'_i, t'_{-i}), t_i) - x_i^{VCG}(t'_i, t'_{-i})] - [v_i(q^\lambda(t_i, t'_{-i}), t_i) - x_i^{VCG}(t_i, t'_{-i})] \\ & = V(q^\lambda(t'_i, t'_{-i}), t_i, t'_{-i}) - V(q^\lambda(t_i, t'_{-i}), t_i, t'_{-i}), \end{aligned}$$

i.e., the ex post gain to misreporting is precisely the ensuing change in social welfare. In the special case when $\lambda_1 = \cdots = \lambda_n$, the difference is nonpositive as $q^\lambda(t)$ solves $\max_c \sum_i v(c, t_i)$. This is the classical VCG result. In general it can be strictly positive and q^λ is not implemented by payments $x^{VCG}(\cdot)$.[5]

In order to implement q^λ we will need some structure and to make matters

[5] To see this, let $N = \{1, 2\}$, $C = \{(c_1, c_2) \in \Re^2_+ : c_1 + c_2 \leq 1\}$, $T_1 = T_2 = [0, 2]$, $v_i(c, t_i) = c_i t_i$, $(\lambda_1, \lambda_2) = (\frac{1}{2}, 1)$ and $t_2 = \frac{1}{2}$. If $\frac{1}{2} < t_1 < 1$ and $t'_1 > 1$, then $q^\lambda(t_1, t_2) = (0, 1)$ and $q^\lambda(t'_1, t_2) = (1, 0)$. We have $U_1(t'_1|t_1, t_2) - U_1(t_1|t_1, t_2) = t_1 - t_2 > 0$.

as simple as possible, we will consider an auction framework. Suppose that the outcome space is $C = \{(c_1, ..., c_n) \in \Re^n_+ : \sum_{i=1}^n c_i \leq 1\}$, and for every i, $T_i = [0, 1]$ and $v_i(c, t_i) = c_i t_i$. Fix a vector $\lambda = (\lambda_1, ..., \lambda_n) \in \Re^n_+$. An explicit form for q^λ can now be derived and it satisfies

$$\begin{aligned} q_i^\lambda(t) &= 1 \text{ if } \lambda_i t_i > \lambda_j t_j \text{ for all } j \neq i, \text{ and} \\ q_i^\lambda(t) &= 0 \text{ if } \lambda_i t_i < \max_{j \neq i} \lambda_j t_j \text{ for some } j. \end{aligned}$$

Note that $t_i \mapsto q_i^\lambda(t_i, t_{-i})$ is nondecreasing for every i and t_{-i}. We will call this property weak individual monotonicity in Section 6. We claim that the payments $x_i(t) = q_i^\lambda(t) \max_{j \neq i} \frac{\lambda_j}{\lambda_i} t_j$ for every i and t implement q^λ and leave the verification to the reader. We will note, however, that if the welfare weights are strictly positive and the same for every agent, then x coincides with the VCG payments x^{VCG} for this auction problem. ■

Example 2 (An auction interdependent values) Consider a standard interdependent value auction with a linear structure as in Bergemann and Morris [2007]. Take $C = \{(c_1, ..., c_n) \in \Re^n_+ : \sum_{i=1}^n c_i \leq 1\}$ and for every i let $v_i(c, t) = c_i(\alpha t_i + \beta \sum_{j \neq i} t_j)$ and $T_i = [0, 1]$. An efficient auction rule is an outcome function $q : T \to C$ and it satisfies

$$\begin{aligned} q_i(t) &= 1 \text{ if } (\alpha - \beta) t_i > (\alpha - \beta) t_j \text{ for all } j \neq i, \text{ and} \\ q_i(t) &= 0 \text{ if } (\alpha - \beta) t_i < (\alpha - \beta) t_j \text{ for some } j. \end{aligned}$$

We claim that payments given by $x_i(t) = q_i(t)[\alpha \max_{j \neq i} t_j + \beta \sum_{k \neq i} t_k]$ for every i and t implement q if two conditions are satisfied: (1) $\alpha > \beta$, which implies that $q_i(\cdot, t_{-i})$ is nondecreasing, and (2) $\alpha > 0$. Note that in the private values special case where $\beta = 0$, $x(\cdot)$ defines the VCG mechanism. We would like to emphasize that in this special case, $q_i(\cdot, t_{-i})$ is still nondecreasing as $\alpha > 0$. If $\alpha < 0$ the VCG mechanism payment for the winning agent i is $x_i^{VCG}(t) = \max_{j \neq i} \alpha t_j \neq \alpha \max_{j \neq i} t_j = x_i(t)$. The VCG mechanism implements the efficient q even when it is not monotone, but x requires monotonicity of q to induce truthful reporting. ■

3 Model

The model consists of the following objects. The set of outcomes is a set $C \subset \Re^m$ and the set of agents is $N = \{1, ..., n\}$. Each agent i has a type space T_i which we take to be the unit interval $[0, 1]$ unless otherwise stated, and a valuation function $v_i : C \times T \to \Re$ where $T = \prod_{j=1}^n T_j$. Hence values are interdependent. As usual collective type $t \in T$ will be denoted (t_i, t_{-i}). Each i is endowed with a partial order $\sqsubseteq_i$ on C and the strict part of $\sqsubseteq_i$ is denoted $\sqsubset_i$. Utilities are quasilinear in money and take the form $u_i(c, x_i, t) = v_i(c, t) - x_i$ where x_i is i's payment. We will maintain the following assumption.

Assumption For every i, c and t_{-i}, $v_i(c, \cdot, t_{-i}) : T_i \to \Re$ is a nondecreasing and continuously differentiable function such that $v_i(c, 0, t_{-i}) = 0$. We denote its derivative at t_i by $v_i'(c, t_i, t_{-i})$.

We are casting the model as one of abstract social choice where social alternatives are m dimensional. By appropriate choice of m and C we can embed several special environments in this model. For example we can analyze auction problems by setting $m = n$ and $C = \{(c_1, ..., c_n) \in \Re^n_+ : \sum_i c_i \leq \bar{q}\}$ for some $\bar{q} > 0$. A host of set allocation problems can also be made a special case and we analyze these problems in depth in Section 6.

An *outcome function* is a map $q : T \to C$. A (direct revelation) *mechanism* is a pair (q, x) where q is an outcome function and $x : T \to \Re^n$ determines payments. We are interested in ex post incentives. A mechanism (q, x) is *ex post incentive compatible* (XIC) if for every i, t_i and t_{-i},

$$t_i \in \arg\max_{t_i' \in T_i} v_i(q(t_i', t_{-i}), t_i, t_{-i}) - x_i(t_i', t_{-i}).$$

If (q, x) is ex post incentive compatible, then truthful reporting constitutes a Nash equilibrium in the complete information games corresponding to every realization of the type vector. A mechanism (q, x) is *ex post individually rational* if for every i and t,

$$v_i(q(t), t) - x_i(t) \geq 0,$$

and *feasible* if $\sum_i x_i(t) \geq 0$ for every i. Individual rationality implies that agents' ex post payoffs at the truth telling equilibrium is at least their outside option which we normalize to zero. Feasibility rules out ex post budget deficits.

An outcome function q is (ex post) *implementable* if there exists a payment rule x such that the mechanism (q, x) is ex post incentive compatible, ex post individually rational and feasible.

4 Monotone Implementation

We are interested in identifying sufficient conditions for implementability of outcome functions. Cremer and McLean [1985] identify such conditions in auction problems when type spaces are finite and completely ordered. We begin by reviewing a modification of their result to fit our abstract social choice environment.

Proposition 1 (cf. Cremer and McLean [1985]) *Suppose that $C = [0, \bar{q}]$ and for every i, $T_i = \{0, 1, ..., \bar{t}_i\}$ for some positive integer $\bar{t}_i$ and $v_i(\cdot, t)$ is nondecreasing. Then $q : T \to C$ is ex post implementable if the following two conditions are satisfied:*

1. *nondecreasing differences*; for every i, t_{-i} $c' < c$ and $t_i \neq \bar{t}_i$,

$$v_i(c, t_i, t_{-i}) - v_i(c', t_i, t_{-i}) \leq v_i(c, t_i + 1, t_{-i}) - v_i(c', t_i + 1, t_{-i}),$$

2. *monotonicity*; for every i, t_{-i} and $t_i \neq \bar{t}_i$,

$$q(t_i, t_{-i}) \leq q(t_i + 1, t_{-i}).$$

The first set of conditions is monotone differences of valuation v_i in (c, t_i) and the second is the monotonicity of the outcome function in t_i. The mechanism used in the proof is "extraction mechanism" in which payments are given for every i and t_{-i} by $x_i^E(0, t_{-i}) = 0$, and

$$x_i^E(t_i, t_{-i}) = \sum\nolimits_{\tau_i=1}^{t_i} [v_i(q(\tau_i, t_{-i}), \tau_i, t_{-i}) - v_i(q(\tau_i - 1, t_{-i}), \tau_i, t_{-i})] \quad (1)$$

for every $t_i \geq 1$. Cremer and McLean use this mechanism in their surplus extraction result with correlated types.

There are two complications that arise in developing a similar result in our model. First, in our framework the outcome space is m-dimensional and partially ordered potentially in a different way by each i. Hence monotonicity of the outcome function needs to be carefully defined. Second,even if we take $m = 1$, $C = [0, \bar{q}]$ and $\sqsubseteq_i$ to be the less than or equal to relation $\leq$ for every i, the Cremer-McLean result still does not apply if type spaces are not finite and completely ordered. An exact extension of the extraction mechanism payments in (1) to the case where type spaces are intervals has not been identified.

In order to overcome these complications we need first to define a useful notion of monotonicity, which will play the key role in our analysis.

Definition 1 An outcome function $q : T \rightarrow C$ is *individually monotone* if for every i and t_{-i}

$$t_i' < t_i \Rightarrow q(t_i, t_{-i}) \sqsubseteq_i q(t_i, t_{-i}).$$

Two cases are of particular interest. If, as in Proposition 1, $m = 1$, $C = [0, \bar{q}]$ and $\sqsubseteq_i$ is the less than or equal to relation on C, then individual monotonicity is precisely condition 2 in Proposition 1. If $m = n$ and $C = \{(c_1, ..., c_n) \in \Re^n_+ : \sum c_j \leq 1\}$, then we may write q in terms of individualized outcome functions $(q_1, ..., q_n)$ where $q_i : T \longrightarrow [0, 1]$. If every i orders C by $c' \sqsubseteq_i c$ if $c'_i \leq c_i$, then individual monotonicity is equivalent to the monotonicity of the maps $q_i(\cdot, t_{-i})$. Thus, individual monotonicity is a suitable extension of familiar monotonicity conditions in the literature.

In order to address the second complication we introduce the class of monotone implementation mechanisms (MIMs).

Definition 2 A mechanism (q, x^M) is a *monotone implementation mechanism* if $t_i \mapsto v'_i(q(t_i, t_{-i}), t_i, t_{-i})$ is Lebesgue integrable for every i and t_{-i}, and

$$x_i^M(t) = v_i(q(t), t) - \int_0^{t_i} v'_i(q(\tau_i, t_{-i}), \tau_i, t_{-i}) d\tau_i. \tag{2}$$

for every i and t.

From now on we will restrict attention to outcome functions which satisfy the integrability condition in Definition 2. Next we show that MIMs can be used to implement individually monotone outcome functions.

Proposition 2 A monotone implementation mechanism (q, x^M) is ex post incentive compatible, ex post individually rational and feasible if the following two conditions are satisfied:

1. nondecreasing differences, i.e., for every i,t_{-i},$t'_i < t_i$ and $c' \sqsubset_i c$

$$v_i(c, t'_i, t_{-i}) - v_i(c', t'_i, t_{-i}) \leq v_i(c, t_i, t_{-i}) - v_i(c', t_i, t_{-i}),$$

2. q is individually monotone.

We would like to make the following remarks regarding Proposition 2.

Remark 1 There is an exact analogy between the conditions identified in Propositions 1 and 2. Not surprisingly, therefore, monotone implementation mechanisms and extraction mechanisms are intimately linked. To see this, consider a single agent problem where $C = [0, 1]$, $T = [0, 1]$ and $v(c, t) = ct$. For any monotone outcome function $q : [0, 1] \to [0, 1]$, the MIM payments are given by $x^M(t) = q(t)t - \int_0^t q(\tau)d\tau$. Fix $\Delta > 0$ such that t/Δ is an integer, let $x^{E,\Delta}(t) = \sum_{k=1}^{t/\Delta}[q(k\Delta) - q((k-1)\Delta)]k\Delta$. Note that $x^{E,\Delta}$ is the payment function for the extraction mechanism when the type space is $\{0, \Delta, 2\Delta, ..., 1\}$. Since q is monotone and the identity map is continuous the Riemann-Stieltjes sum $\sum_{k=1}^{t/\Delta}[q(k\Delta) - q((k-1)\Delta)]k\Delta$ converges to Riemann-Stieltjes integral $\int_0^t \tau dq(\tau)$ as Δ converges to zero. Integrating by parts we get $\int_0^t \tau dq(\tau) = tq(t) - \int_0^t q(\tau)d\tau = x^M(t)$. Hence $x^{E,\Delta}$ approximates x^M for small Δ.

Remark 2 The definitions of individual monotonicity and nondecreasing differences may be adapted to let the partial order $\sqsubseteq_i$ to depend on the type profile t_{-i} of the rest of the agents and Proposition 2 would still hold.

Proposition 2 has obvious implications for the two examples of Section 2.

Example 1 continued In the weighted welfare maximization example, the monotone implementation mechanism payments associated with q^λ can be calculated. Suppose that $t_i > \max_{j \neq i} \frac{\lambda_j}{\lambda_i} t_j$. Then $q_i^\lambda(t) = 1$ and

$$x_i^M(t) = t_i - \int_0^{t_i} q_i^\lambda(\tau_i, t_{-i})d\tau_i = t_i - \int_{\max_{j\neq i} \frac{\lambda_j}{\lambda_i} t_j}^{t_i} d\tau_i = \max_{j \neq i} \frac{\lambda_j}{\lambda_i} t_j.$$

If $t_i = \max_{j \neq i} \frac{\lambda_j}{\lambda_i} t_j$, then $x_i^M(t) = q_i^\lambda(t) \max_{j \neq i} \frac{\lambda_j}{\lambda_i} t_j$ and if $t_i < \max_{j \neq i} \frac{\lambda_j}{\lambda_i} t_j$, then $x_i^M(t) = 0$. In all three cases $x_i^M(t) = q_i^\lambda(t) \max_{j \neq i} \frac{\lambda_j}{\lambda_i} t_j$. Since q^λ is individually monotone with respect to $\leq$ for every i and since $v_i(c, t_i) = c_i t_i$ satisfies nondecreasing differences, q^λ is ex post implemented by x^M. ■

Example 2 continued In the efficient auction example, the monotone implementation mechanism payments can be calculated as follows. Suppose that $\alpha > \max\{\beta, 0\}$. If $t_i > \max_{j \neq i} t_j$, then $q_i(t) = 1$ and

$$x_i^M(t) = \alpha t_i + \beta \sum_{j \neq i} t_j - \int_{\max_{j \neq i} t_j}^{t_i} \alpha d\tau_i = \alpha \max_{j \neq i} t_j + \beta \sum_{j \neq i} t_j.$$

If $t_i = \max_{j \neq i} t_j$, then $x_i^M(t) = q_i(t)[\alpha t_i + \beta \sum_{j \neq i} t_j] = q_i(t)[\alpha \max_{j \neq i} t_j + \beta \sum_{j \neq i} t_j]$, and if $t_i < \max_{j \neq i} t_j$, then $x_i^M(t) = 0$. In all three cases, then, $x_i^M(t) = q_i(t)[\alpha \max_{j \neq i} t_j + \beta \sum_{j \neq i} t_j]$. Note that since $\alpha > \max\{\beta, 0\}$, q satisfies individual monotonicity and valuations satisfy nondecreasing differences. We conclude by Proposition 2 that q is ex post implemented by x^M. ■

5 Efficient Implementation

In light of Propositions 1 and 2, in order to determine ex post implementability of outcome functions we need only identify conditions under which they are individually monotone. Of particular interest are efficient outcome functions.

Let $V(c, t) = \sum_{j=1}^{n} v_j(c, t)$ denote the social welfare at outcome c if the type profile is t.

Definition 3 An outcome function $q : T \to C$ is (ex post) *efficient* if for every t, $q(t)$ solves

$$\max_{c \in C} V(c, t). \tag{3}$$

Now we can identify a set of conditions on the outcome space C and social welfare function V under which efficient outcome functions are individually monotone using the theory of monotone optimal solutions.

Proposition 3 An efficient outcome function q satisfies individual monotonicity if the following conditions are satisfied:

1. for every i, C is a lattice when ordered with $\sqsubseteq_i$,

2. for every i and t, $V(\cdot, t) : (C, \sqsubseteq_i) \rightarrow \Re$ is supermodular, and

3. for every i and t_{-i}, $V : (\cdot, \cdot, t_{-i}) : (C, \sqsubseteq_i) \times T_i \rightarrow \Re$ satisfies strict single crossing in (c, t_i).

The proof of Proposition 3 is a straightforward application of the theory of increasing optimal solutions. For every i and t_{-i}, (3) satisfies the conditions in Theorem 2.8.7 in Topkis [1998]. In particular, the constraint set is a lattice, the set of parameters t_i is linearly ordered, the objective is supermodular in the choice variable c and it satisfies strict single crossing in (c, t_i). Thus, fixing i and t_{-i}, we conclude that any selection of solutions $q(\cdot, t_{-i})$ to (4) must be nondecreasing.

Combining Propositions 2 and 3, (or Propositions 1 and 3) we conclude that an efficient outcome function is ex post implementable if the three conditions of Proposition 3 are satisfied and valuations satisfy the nondecreasing differences property in Propositions 1 and 2. We record this observation next.

Corollary 1 An efficient outcome function $q : T \rightarrow C$ is implementable if the following conditions are satisfied:

1. for every i and t_{-i}, $v_i(\cdot, \cdot, t_{-i}) : (C, \sqsubseteq_i) \times T_i \rightarrow \Re$ satisfies nondecreasing differences

2. for every i, C is a lattice when ordered with $\sqsubseteq_i$,

3. for every i and t, $V(\cdot, t) : (C, \sqsubseteq_i) \rightarrow \Re$ is supermodular, and

4. for every i and t_{-i}, $V : (\cdot, \cdot, t_{-i}) : (C, \sqsubseteq_i) \times T_i \rightarrow \Re$ satisfies strict single crossing in (c, t_i).

There are important environments in which the outcome set is linearly ordered. If this is the case, the lattice and supermodularity conditions are automatically satisfied and one need only check the strict single crossing properties to apply Corollary 1.

An important example is the linear social choice environment in Jehiel and Moldovanu [2001]. In their model the set of outcomes is an arbitrary finite set $C = \{c_1, ..., c_K\}$. Valuations are linear in types and are given by $v_i(c,t) = \sum_{j=1}^{n} a_{ij}(c)t_j$, where the maps $a_{ij}(\cdot) : C \to \Re$ are common knowledge. The coefficient $a_{ij}(c)$ measures the marginal effect of j's private information on i's value for outcome c. Jehiel and Moldovanu conveniently order C for every i by $c' \sqsubseteq_i c$ if $a_{ii}(c') \leq a_{ii}(c)$. This implies that valuations satisfy the nondecreasing differences property in Propositions 1-3 and Corollary 1. As another consequence, $(C, \sqsubseteq_i)$ is completely ordered for every i. All that remains to show, then, is that V satisfies the strict single crossing properties. Jehiel and Moldovanu introduce the "weak congruence" condition

$$a_{ii}(c') < a_{ii}(c) \Rightarrow \sum_j a_{ji}(c') < \sum_j a_{ji}(c) \tag{4}$$

exactly for this purpose. Note that in this linear model, $V(c,t) = \sum_i \sum_j a_{ij}(c)t_j$ and $\frac{\partial V}{\partial t_i}(c,t) = \sum_j a_{ji}(c)$. As a result $V(\cdot,\cdot,t_{-i}) : (C, \sqsubseteq_i) \times T_i \to \Re$ satisfies strictly increasing differences if $c' \sqsubset_i c$ implies $\sum_j a_{ji}(c') < \sum_j a_{ji}(c)$. By definition of $\sqsubseteq_i$, this statement is equivalent to (4). Since strictly increasing differences is stronger than strict single crossing property, the implementability of the efficient outcome function in Jehiel and Moldovanu can be obtained using our Corollary 1.

Bergemann and Valimaki [2002] analyze a nonlinear version of the Jehiel-Moldovanu environment and show that if q is efficient and individually monotone, and if valuations satisfy nondecreasing differences in outcomes and agents' own types, then it is implementable. Since their outcome set is completely ordered (for every i and t_{-i}) Proposition 2 extends their result to partially ordered environments and to monotone outcome functions that need not be efficient.

In many problems social outcomes do not form a lattice and Proposition 3 and Corollary 1 do not apply. We will develop an approach for such problems within a more structured framework in the next section, concentrating on social outcomes which specify a set for every agent.

6 Set Allocation Problems

In this section we will analyze mechanism design problems in which social outcomes involve specification of a set for each agent. A *set allocation problem* (with no consumption externalities) consists of the following ingredients. For some finite set Ω, the set of social outcomes is $C \subseteq 2^{\Omega} \times \cdots \times 2^{\Omega}$ (n times). If $c \in C$, we will sometimes switch the order of coordinates and write $c = (c_i, c_{-i})$. For every i, t, c and c', $v_i(c,t) = v_i(c',t)$ if $c_i = c'_i$. We will denote the projection of C onto its ith coordinate by $C_i \subseteq 2^{\Omega}$. Abusing notation, we will write $v_i(c,t) = v_i(c_i,t)$ and treat v_i to be a map defined on $C_i \times T$. Hence we continue allowing for informational externalities but there are no externalities pertaining to the allocation.

The order structure is as follows. Each i uses the componentwise extension of the weak inclusion order $\subseteq$ on C, in other words, for every i, $c' \sqsubseteq_i c$ if and only if $c'_j \subseteq c_j$ for every j. We will denote this order by $\subseteq^n$. The strict order $\subsetneq^n$ is defined by $c'_j \subseteq c_j$ for every j and $c'_j \subset c_j$ for some j.

In order to embed this environment in the framework of the last two sections, set $m = |\Omega| n$, and write $\tilde{C} = \{(\chi_{c_1}, ..., \chi_{c_n}) : (c_1, ..., c_n) \in C\}$. Every social outcome in C is identifiable by an outcome in $\tilde{C}$ and vice versa. If we appropriately change the definition of nondecreasing differences to account for the restricted domain $C_i \times T$, then the results in the previous sections apply to problems in which C is the set of outcomes.

The set allocation environment reduces to a number of interesting applications via appropriate choice of the sets Ω and C.

Mergers Let $\Omega = N$ and $C = \{(c_1, ..., c_n) : i \in c_i$ for every i and $c_i = c_j$ if $j \in c_i\}$. The set c_i allocated to i is interpreted to be the set of agents with whom i is merged. Note that i always merges with himself and if i merges with j, i also merges with every other agent with whom j merges. Importantly, $(C, \subseteq^n)$ is a lattice and Proposition 3 renders the efficient merger rule individually monotone if supermodularity and strict single crossing properties are satisfied. ■

Combinatorial allocation Let Ω be a set of objects and $C = \{(c_1, ..., c_n) : \cup_i c_i \subseteq \Omega$ and $c_i \cap c_j = \varnothing$ if $i \neq j\}$. The set c_i is now interpreted as the set of objects i receives. Note that $(C, \subseteq^n)$ is not a lattice and Proposition 3 does not apply. ■

Other problems which can be analyzed using a set allocation framework include public good provision problems and queueing problems.

If C has the separable structure $\prod_i C_i$, as it does in set allocation problems, then an outcome function $q : T \to C$ can be written in terms of individualized outcome functions $(q_1, ..., q_n)$ where $q_i : T \to C_i$. Implementability and efficiency of q are defined exactly as before. With the current order structure, an outcome function $q : T \to C$ is individually monotone if for every i, t_{-i} and $t'_i < t_i$, we have $q(t'_i, t_{-i}) \subseteq^n q(t_i, t_{-i})$, i.e., we have $q_j(t'_i, t_{-i}) \subseteq q_j(t_i, t_{-i})$ for every $j = 1, ..., n$. In other words, q is individually monotone if the set allocated to every agent enlarges if one agent's type goes up.

An exact analog of Proposition 2 applies in set allocation problems, which indicates that an outcome function q is implementable if q is individually monotone and if for every i and t_{-i}, $v_i(\cdot, \cdot, t_{-i}) : (C_i, \subseteq) \times T_i \to \Re$ satisfies nondecreasing differences. We will now show that the notion of monotonicity can be weakened.

Definition 4 In a set allocation problem, an outcome function $q = (q_1, ..., q_n)$

satisfies *weak individual monotonicity* if for every i and t_{-i}, $t_i' < t_i$ implies $q_i(t_i', t_{-i}) \subseteq q_i(t_i, t_{-i})$.

Thus if q satisfies individual monotonicity, then only i's set is required to expand as a result of a change in i's type. This is certainly weaker than the requirement that all sets should expand as a result, as in individual monotonicity. Now we can prove the following result on sufficient conditions for implementability.

Proposition 4 An outcome function q in a set allocation problem is implementable if

1. q satisfies weak individual monotonicity, and
2. for every i and t_{-i}, $v_i(\cdot, \cdot, t_{-i}) : (C_i, \subseteq) \times T_i \to \Re$ satisfies nondecreasing differences.

Note that the interdependent value auction example in Section 2 is a fluid extension of the current framework. The efficient auction rule is implementable precisely because under the condition $\alpha > \max\{\beta, 0\}$ it satisfies weak individual monotonicity and valuations satisfy nondecreasing differences over $[0, 1] \times T_i$.

Next we will identify a set of sufficient conditions for efficient outcome functions to satisfy individual monotonicity in the set allocation environment. If the social outcomes form a lattice, then such conditions are on individual valuations rather than the social welfare function. First we note the following fact.

Lemma 1 If $(C, \subseteq^n)$ is a lattice then $(C_i, \subseteq)$ is a sublattice of $(2^\Omega, \subseteq)$ for every i.

Consequently, if the set of social outcomes C is a lattice, we can meaningfully impose the condition that an individual valuation v_i is supermodular on C_i for every t. This condition implies a complementarity relationship between elements of Ω.

Proposition 5 *Let $q = (q_1, ..., q_n)$ be an efficient outcome function in the set allocation problem. Then q satisfies individual monotonicity if the following conditions are satisfied:*

1. $(C, \subseteq^n)$ *is a lattice*

2. *for every i and t, $v_i(\cdot, t) : (C_i, \subseteq) \rightarrow \Re$ is supermodular, and*

3. *for every i, j and t_{-i}, $v_j(\cdot, \cdot, t_{-i}) : (C_j, \subseteq) \times T_j \rightarrow \Re$ satisfies strictly increasing differences.*

The condition that the outcome set is a lattice when ordered with the componentwise extension of $\subseteq$ is satisfied in certain interesting applications and violated in others. In public good problems and in merger problems this condition is satisfied. In auction problems it fails. To develop an approach to address the environments in which the outcome set is not a lattice, define $C_{-i}(c_i) = \{c_{-i} : (c_i, c_{-i}) \in C\}$ and

$$v^*_{-i}(c_i, t) = \max_{c_{-i} \in C_{-i}(c_i)} \textstyle\sum_{j \neq i} v_j(c_j, t).$$

as the largest surplus attainable by the rest of the agents at type profile t, if i is allocated set c_i. Next define

$$v^*_i(c_i, t) = v_i(c_i, t) + v^*_{-i}(c_i, t)$$

and note that $q(t)$ solves $\max_{c \in C} V(c, t)$ if and only if for every i, $q_i(t)$ solves

$$\max_{c_i \in C_i} v^*_i(c_i, t). \tag{5}$$

If C_i has a lattice structure, then (5) allows us to use the theory of monotone selection since its constraint set is a lattice. This observation leads to the following result.

Proposition 6 Let $q = (q_1, ..., q_n)$ be an efficient outcome function in a set allocation problem. Then q satisfies weak individual monotonicity if the following conditions are satisfied.

1. $(C_i, \subseteq)$ is a sublattice of $(2^\Omega, \subseteq)$ for every i,
2. $v_i^*(\cdot, t) : (C_i, \subseteq) \to \Re$ is supermodular for every t, and
3. $v_i^*(\cdot, \cdot, t_{-i}) : (C_i, \subseteq) \times T_i \to \Re$ satisfies strict single crossing property for every i and t_{-i}.

The supermodularity and the strict single crossing conditions in Proposition 6 require further comment. The strict single crossing property of v_i^* in (c_i, t_i) is equivalent to the statement

$$\begin{array}{rcl} v_i(c_i, t_i', t_{-i}) - v_i(c_i', t_i', t_{-i}) & \geq & v_{-i}^*(c_i', t_i', t_{-i}) - v_{-i}^*(c_i, t_i', t_{-i}) \\ & \Rightarrow & \\ v_i(c_i, t_i, t_{-i}) - v_i(c_i', t_i, t_{-i}) & > & v_{-i}^*(c_i', t_i, t_{-i}) - v_{-i}^*(c_i, t_i, t_{-i}) \end{array}$$

for every $i, t_{-i}, t_i' < t_i$ and $c_i' \subset c_i$. Let us first note that in the case of private values, the right hand sides of the both inequalities are equal. As a result, the extended strict single crossing property is implied by strictly increasing differences of individual valuation functions. In general, the strict single crossing property captures the informal idea that t_i should have a larger effect on the marginal valuations of i than it does on the marginal valuations of $j \neq i$. Such an assumption seems critical to proving individual monotonicity of ex post outcome efficient mechanisms. In specialized settings, an interagent crossing condition is sufficient. To see this let $\Omega = \{\omega\}$ and $v_i(c_i, t) = \mu_i(c_i) w_i(t)$ where w_i is differentiable and $0 \leq \mu_i(\varnothing) < \mu_i(\Omega)$. Now the extended strict single crossing property follows if $\frac{\partial w_i}{\partial t_i}(t) > \frac{\partial w_j}{\partial t_i}(t)$ for every i and t. This is precisely the condition that t_i has a larger effect on i's marginal value for the outcome variable. Similar assumptions designed for the same purpose have appeared in all work on auctions with interdependent values, e.g., Cremer and McLean [1985], Ausubel [1999], Dasgupta and Maskin [2001] and Perry and Reny [2002].

The supermodularity condition is satisfied in some but not all cases of interest.

Example 3 (Combinatorial allocation with two agents) Consider a two-agent combinatorial allocation problem where $N = \{1, 2\}$ and $C = \{(c_1, c_2) : c_i \subseteq \Omega$ and $c_1 \cap c_2 = \varnothing\}$ for some finite set Ω. Then $c_2 \mapsto v_1^*(c_2, t)$ is supermodular on C_2 and $c_1 \mapsto v_2^*(c_1, t)$ is supermodular on C_1. To see this, note that

$$C_2(c_1') = 2^{\Omega \backslash c_1'} \text{ and } C_2(c_1'') = 2^{\Omega \backslash c_1''}$$

Let

$$c_2' \in \arg \max_{c_2 \in C_2(c_1')} v_2(c_2, t) \text{ and } c_2'' \in \arg \max_{c_2 \in C_2(c_1'')} v_2(c_2, t).$$

It follows that

$$\begin{aligned} c_2' \cup c_2'' &\subseteq \Omega \backslash c_1' \cup \Omega \backslash c_1'' = \Omega \backslash (c_1' \cup c_1''), \text{ and} \\ c_2' \cap c_2'' &\subseteq \Omega \backslash c_1' \cap \Omega \backslash c_1'' = \Omega \backslash (c_1' \cap c_1'') \end{aligned}$$

Since $c_2' \cup c_2'' \in C_2$ and $c_2' \cap c_2'' \in C_2$, we conclude that

$$\begin{aligned} v_2^*(c_1', t) + v_2^*(c_1'', t) &= v_2(c_2', t) + v_2(c_2'', t) \\ &\leq v_2(c_2' \cup c_2'', t) + v_2(c_2' \cap c_2'', t) \\ &\leq v_2^*(c_1' \cup c_1'', t) + v_2^*(c_1' \cap c_1'', t) \end{aligned}$$

where the first inequality follows from the supermodularity of $v_2(\cdot, t)$. Similarly for $v_1^*(\cdot, t)$. ■

Example 4 (Allocating 2 objects) Consider the problem of allocating two indivisible objects to n agents. Let $N = \{1, ..., n\}$ and $\Omega = \{\omega_1, \omega_2\}$ and set $C = \{(c_1, ..., c_n) : c_i \subseteq \Omega \text{ and } c_i \cap c_j = \varnothing \text{ if } i \neq j\}$. The problem of finding the revenue-maximizing mechanism in this setting was considered by Levin [1997]. In this very simple case, $C_i = 2^\Omega$ for each i and $c_i \mapsto v^*_{-i}(c_i, t)$ is supermodular if and only if

$$v^*_{-i}(\{\omega_1\}, t) + v^*_{-i}(\{\omega_2\}, t) \leq v^*_{-i}(\varnothing, t) + v^*_{-i}(\{\omega_1, \omega_2\}, t) = v^*_{-i}(\varnothing, t).$$

Suppose that $v_j(\{\omega_1\}, t) = v^*_{-i}(\{\omega_2\}, t)$ and $v_k(\{\omega_2\}, t) = v^*_{-i}(\{\omega_1\}, t)$. If $j = k$, then supermodularity of $v_k(\cdot, t)$ implies that

$$\begin{aligned} v^*_{-i}(\{\omega_2\}, t) + v^*_{-i}(\{\omega_1\}, t) &= v_j(\{\omega_1\}, t) + v_j(\{\omega_2\}, t) \\ &\leq v_j(\{\omega_1, \omega_2\}, t) + v_j(\varnothing, t) \\ &= v^*_{-i}(\{\omega_1, \omega_2\}, t). \end{aligned}$$

If $j \neq k$, then the allocation where $c_j = \{\omega_1\}, c_k = \{\omega_2\}$ and $c_q = \varnothing$ for all other $q \neq i, j, k$ is feasible for the problem $\max_{c_{-i} \in C_{-i}(\varnothing)} \sum_{j \neq i} v_j(c_j, t)$ from which we deduce that

$$v^*_{-i}(\{\omega_2\}, t) + v^*_{-i}(\{\omega_1\}, t) = v_j(\{\omega_1\}, t) + v_k(\{\omega_2\}, t) \leq v^*_{-i}(\varnothing, t)$$

and $c_i \mapsto v^*_{-i}(c_i, t)$ is supermodular. ■

The argument used in Example 4 is specific to the case of two indivisible objects and breaks down with more that two objects. Unfortunately, the function $v^*_{-i}(\cdot, t)$ will not generally be supermodular on C_i when C is not a lattice. On the other hand, there are applications involving valuation functions whose special structure does not require supermodularity conditions.

Example 5 (Multiunit allocation problems) Consider a multiunit allocation problem in which m units of of an indivisible object will be allocated between the agents. In this application agents are only interested in the number of units they receive. It is therefore useful to write $\hat{v}_i(|c_i|, t) = v_i(c_i, t)$ and transform the outcome space to $C' = \{(k_1, ..., k_n) \in \mathbb{Z}_+^n : \sum_i k_i \leq m\}$. As a result, we have

$$\begin{aligned} C_i &= \{0, 1, ..., m\}, \\ C_{-i} &= \{k_{-i} \in \mathbb{Z}_+^{n-1} : \sum\nolimits_{j \neq i} j_i \leq m\}, \text{ and} \\ C_{-i}(c_i) &= \{k_{-i} \in \mathbb{Z}_+^{n-1} : \sum\nolimits_{j \neq i} k_i \leq m - k_i\}. \end{aligned}$$

Suppose that there exist maps $w_i : L_i \times T \to \Re_+$ such that

$$\hat{v}_i(k_i, t) = \sum\nolimits_{k=1}^{k_i} w_i(k, t).$$

The number $w_i(k, t)$ is the marginal valuation of agent i for the kth unit, if the type vector is t. Suppose that $w_i(0, t) = 0$. In this environment, the lattice and supermodularity conditions are automatically satisfied and an efficient outcome function is weakly individually monotone if the extended strict single crossing property is satisfied. The payments of the monotone implementation mechanism which implements the efficient outcome function can be explicitly calculated. Fix a weakly individually monotone allocation rule q, an agent i, and a type vector t. Define $\tau_i^k(t_{-i}) = \inf\{z : q_i(y, t_{-i}) = k\}$ for $k = 1, ..., q_i(t)$ and let $\tau^{k+1} = t_i$. In words, $\tau_i^k(t_{-i})$ is agent i's critical type for the kth unit. We have

$$\begin{aligned} x_i(t) &= \sum\nolimits_{k=1}^{q_i(t)} w_i(k, t) - \int_0^{t_i} \sum\nolimits_{k=1}^{q_i(y, t_{-i})} w_i'(k, y, t_{-i}) dy \\ &= \sum\nolimits_{k=1}^{q_i(t)} w_i(k, t) - \sum\nolimits_{l=1}^{q_i(t)} \int_{\tau_i^l(t_{-i})}^{\tau_i^{l+1}(t_{-i})} \sum\nolimits_{k=1}^{l} w_i'(k, y, t_{-i}) dy \\ &= \sum\nolimits_{k=1}^{q_i(t)} w_i(k, \tau_i^k(t_{-i}), t_{-i}). \end{aligned}$$

Thus, in a multiunit model with an indivisible object MI mechanism asks each agent to pay the sum of marginal valuations for each unit that he obtains, where these marginals are evaluated at his critical type for that unit. Analogous results have been obtained in Ausubel (1999) in his analysis of the generalization of the Vickrey auction to an interdependent value environment.■

7 Conclusion

We analyzed monotone implementation in a fairly large class of mechanism design environments. Of critical importance to our analysis was the assumption that type spaces are intervals. However this seems to be an indispensable assumption in mechanism design with interdependent values and ex post incentive constraints. By partially ordering outcome spaces we can identify a class of outcome functions which can be implemented in ex post equilibrium. By making lattice, supermodularity and single crossing assumptions we can apply the theory of monotone selection of optimizers and conclude that efficient social outcomes are implementable. Many ad hoc outcome functions (which are not necessarily solutions to a mechanism design problem) satisfy individual monotonicity and are therefore implementable.

The close connection between the Cremer-McLean mechanisms with finitely many types and the monotone implementation mechanisms when type spaces are intervals raise an intriguing question regarding whether or not full surplus extraction results can be obtained in our framework. We leave this question for future research.

8 Appendix: Proofs

Proof of Proposition 1 Define payments $x^E : T \to \Re^n$ as in the main body of the text, by $x_i^E(0, t_{-i}) = 0$, and

$$x_i^E(t_i, t_{-i}) = \sum_{\tau_i=1}^{t_i} [v_i(q(\tau_i, t_{-i}), \tau_i, t_{-i}) - v_i(q(\tau_i - 1, t_{-i}), \tau_i, t_{-i})]$$

for every $t_i \geq 1$. We will show that if q is monotone and v_i satisfies nondecreasing differences in (c, t_i), then (q, x^E) is ex post incentive compatible.

Fix i, t_{-i} and $t_i' < t_i$ and note that $q(t_i', t_{-i}) \leq q(t_i, t_{-i})$. By nondecreasing differences, we get

$$\begin{aligned} & \sum_{\tau_i=t_i'+1}^{t_i} [v_i(q(\tau_i, t_{-i}), \tau_i, t_{-i}) - v_i(q(\tau_i - 1, t_{-i}), \tau_i, t_{-i})] \\ \leq & \sum_{\tau_i=t_i'+1}^{t_i} [v_i(q(\tau_i, t_{-i}), t_i, t_{-i}) - v_i(q(\tau_i - 1, t_{-i}), t_i, t_{-i})] \\ = & \ v_i(q(t_i, t_{-i}), t_i, t_{-i}) - v_i(q(t_i', t_{-i}), t_i, t_{-i}). \end{aligned}$$

Therefore,

$$\begin{aligned} v_i(q(t_i, t_{-i}), t_i, t_{-i}) - x_i^E(t_i, t_{-i}) &= v_i(q(t_i, t_{-i}), t_i, t_{-i}) \\ &\quad - \sum_{\tau_i=1}^{t_i} [v_i(q(\tau_i, t_{-i}), \tau_i, t_{-i}) - v_i(q(\tau_i - 1, t_{-i}), \tau_i, t_{-i})] \\ &= v_i(q(t_i, t_{-i}), t_i, t_{-i}) \\ &\quad - \sum_{\tau_i=1}^{t_i'} [v_i(q(\tau_i, t_{-i}), \tau_i, t_{-i}) - v_i(q(\tau_i - 1, t_{-i}), \tau_i, t_{-i})] \\ &\quad - \sum_{\tau_i=t_i'+1}^{t_i} [v_i(q(\tau_i, t_{-i}), \tau_i, t_{-i}) - v_i(q(\tau_i - 1, t_{-i}), \tau_i, t_{-i})] \\ &\geq v_i(q(t_i, t_{-i}), t_i, t_{-i}) \\ &\quad - \sum_{\tau_i=1}^{t_i'} [v_i(q(\tau_i, t_{-i}), \tau_i, t_{-i}) - v_i(q(\tau_i - 1, t_{-i}), \tau_i, t_{-i})] \\ &\quad - [v_i(q(t_i, t_{-i}), t_i, t_{-i}) - v_i(q(t_i', t_{-i}), t_i, t_{-i})] \\ &= v_i(q(t_i', t_{-i}), t_i, t_{-i}) - x_i^E(t_i', t_{-i}). \end{aligned}$$

A similar argument applies if $t_i' > t_i$. Thus (q, x^E) is ex post incentive compatible. ■

Proof of Proposition 2 We will show that if x is as defined in (1), then (q, x) is ex post incentive compatible, ex post individually rational and feasible. Fix i, t_{-i} and $t_i' < t_i$ and an individually monotone outcome function q. We have

$$\begin{aligned} x_i(t_i, t_{-i}) - x_i(t_i', t_{-i}) &= v_i(q(t_i, t_{-i}), t_i, t_{-i}) - v_i(q(t_i', t_{-i}), t_i', t_{-i}) \\ &\quad - \int_{t_i'}^{t_i} v_i'(q(z, t_{-i}), z, t_{-i}) dz \\ &\leq v_i(q(t_i, t_{-i}), t_i, t_{-i}) - v(q_i(t_i', t_{-i}), t_i', t_{-i}) \\ &\quad - \int_{t_i'}^{t_i} v_i'(q(t_i', t_{-i}), z, t_{-i}) dz \\ &= v_i(q(t_i, t_{-i}), t_i, t_{-i}) - v_i(q(t_i', t_{-i}), t_i, t_{-i}) \end{aligned}$$

where the first equality follows from (1), the inequality from conditions 1 and 2 in the proposition, and the ultimate equality follows from the integrability of $v_i'(c, \cdot, t_{-i})$. Similarly if $t_i < t_i'$. Thus the monotone implementation mechanism (q, x) is ex post incentive compatible. To check ex post individual rationality, note that

$$v_i(q(t), t) - x_i(t) = \int_0^{t_i} v_i'(q(z, t_{-i}), z, t_{-i}) dz \geq 0$$

since $v_i'(c, z, t_{-i}) \geq 0$ for every c, t_{-i} and z. Finally, we have

$$\begin{aligned} \int_0^{t_i} v_i'(q(z, t_{-i}), z, t_{-i}) dz &\leq \int_0^{t_i} v_i'(q(t_i, t_{-i}), z, t_{-i}) dz \\ &= v_i(q(t_i, t_{-i}), t_i, t_{-i}) \end{aligned}$$

where we use $v_i(q(t_i, t_{-i}), 0, t_{-i}) = 0$. It follows that $x_i(t) \geq 0$ and (q, x) is feasible. This completes the proof. ■

Proof of Proposition 3 See the text for an informal argument, which can easily be formalized.■

Proof of Proposition 4 The argument mimics closely the proof of Proposition 2.■

Proof of Proposition 5 In light of Proposition 3, it suffices to show that $V(\cdot, t) : (C, \subseteq^n)$ is supermodular for every t, and $V(\cdot, \cdot, t_{-i}) : (C, \subseteq^n) \times T_i \to \Re$ satisfies strict single crossing property for every i and t_{-i}.

To show that $V(\cdot, t)$ is supermodular, pick c and c' and note that

$$\begin{aligned} V(c,t) + V(c',t) &= \textstyle\sum_j (v_j(c_j, t) + v_j(c'_j, t)) \\ &= \textstyle\sum_j (v_j(c'_j \cup c_j, t) + v_j(c'_j \cap c_j, t)) \\ &= V(c \vee c') + V(c \wedge c'). \end{aligned}$$

We will show that $V(\cdot, \cdot, t_{-i})$ satisfies strictly increasing differences, which is a stronger condition than strict single crossing. Pick $c' \subset^n c$ and $t'_i < t_i$. Note that $c'_j \subseteq c_j$ for every j and $c'_j \subset c_j$ for some j. We have

$$\begin{aligned} V(c, t'_i, t_{-i}) - V(c', t'_i, t_{-i}) &= \textstyle\sum_j (v_j(c_j, t'_i, t_{-i}) - v_j(c'_j, t'_i, t_{-i})) \\ &< \textstyle\sum_j (v_j(c_j, t_i, t_{-i}) - v_j(c'_j, t_i, t_{-i})) \\ &= V(c, t_i, t_{-i}) - V(c', t_i, t_{-i}) \end{aligned}$$

and the proof is complete.■

Proof of Proposition 6 The proof follows from a straightforward application of Theorem 2.8.7 in Topkis [1998] in Problem (5).■

Private values We will now argue that monotone implementation mechanisms constitute a suitable extension of VCG mechanisms to interdependent value environments by showing that the two mechanisms coincide under private values. We will establish this in a combinatorial allocation problem. Under suitable condition the result holds in all social choice problems.

Suppose that values are private. A *VCG mechanism* is a pair (q, x^{VCG}) where f is ex post efficient and x^{VCG} is such that for every i and t,

$$x_i^{VCG}(t) = \max_{c_{-i} \in C_{-i}} \sum\nolimits_{j \neq i} v_j(c_j, t_j) - \max_{c_{-i} \in A_{-i}(f_i(t))} \sum_{j \neq i} v_j(a_j, t_j). \tag{6}$$

Proposition 7 Suppose that values are private and that q is an efficient and weakly individually monotone. If (q, x^M) is a monotone implementation mechanism and (q, x^{VCG}) is a VCG mechanism, then $x^M = x^{VCG}$.

Proof. Fix i and t_{-i} and note that $\{q_i(z, t_{-i}) : z \leq t_i\}$ is a completely ordered (by $\subseteq$) subset of 2^Ω. Write $\{q_i(z, t_{-i}) : z \leq t_i\} = \{c_i^1, ..., c_i^l\}$ where $c_i^k \subseteq c_i^{k+1}$ for every $k = 1, ..., l-1$. Divide $[0, 1]$ into subsets $\Gamma_i^1, ..., \Gamma_i^l$ such that $q_i(z, t_{-i}) = c_i^k$ if and only if $z \in \Gamma_i^k$. Note that each Γ_i^k is nonempty and connected, and that $[0_i, t_i] = \cup_{k=1}^l \Gamma_i^k$. Consequently $\inf \Gamma_i^k = \sup \Gamma_i^{k-1}$. Let $\tau_i^k = \inf \Gamma_i^k$. For every $k \neq 1$, by standard continuity arguments we must have

$$V((c_i^k, q_{-i}(\tau_i^k, t_{-i})), \tau_i^k, t_{-i}) = V((c_i^{k-1}, q_{-i}(\tau_i^{k-1}, t_{-i})), \tau_i^k, t_{-i})$$

implying

$$v_i(c_i^k, \tau_i^k) - v_i(c_i^{k-1}, \tau_i^k) = \max_{c_{-i} \in C_{-i}(c_i^{k-1})} \sum_{j \neq i} v_j(c_j, t_j) - \max_{ac \in C_{-i}(c_i^k)} \sum_{j \neq i} v_j(c_j, t_j) \tag{7}$$

Now we can write, by letting $\tau_i^{m+1} = t_i$ and $\tau_i^0 = 0$

$$
\begin{aligned}
x_i^M(t) &= v_i(q_i(t), t_i) - \int_0^{t_i} v_i'(q_i(z, t_{-i}), z)dz \\
&= v_i(c_i^l, t_i) - \sum_{k=1}^{l} \int_{\tau_i^k}^{\tau_i^{k+1}} v_i'(c_i^k, z)dz \\
&= \sum_{k=1}^{l} [v_i(c_i^k, \tau_i^k) - v_i(c_i^{k-1}, \tau_i^k)] \\
&= \sum_{k=1}^{l} \left[\max_{c_{-i} \in C_{-i}(c_i^{k-1})} \sum_{j \neq i} v_j(c_j, t_j) - \max_{c_{-i} \in C_{-i}(c_i^k)} \sum_{j \neq i} v_j(a_j, t_j) \right] \\
&= \max_{c_{-i} \in C_{-i}(\varnothing)} \sum_{j \neq i} v_j(c_j, t_j) - \max_{c_{-i} \in C_{-i}(c_i^l)} \sum_{j \neq i} v_j(c_j, t_j) \\
&= x_i^{VCG}(t).
\end{aligned}
$$

and this completes the proof. ■

Exactly the same argument would show that VCG mechanisms and monotone implementation mechanisms coincide in more general environments with private values if the following conditions are satisfied (1) efficient outcome functions are individually monotone, (2) for every i $(\inf_i C_i)_{i \in N} \in A$ and (3) $C_{-i}(\inf_i C_i) = C_{-i}$.

References

[1] L. Ausubel, A Generalized Vickrey Auction, University of Maryland working paper, 1999.

[2] D. Bergemann and S. Morris, An Ascending Auction for Interdependent Values: Uniqueness and Robustness to Strategic Uncertainty, American Economic Review Papers and Proceedings, 97:2, 2007, 125-130.

[3] D. Bergemann and J. Valimaki, Information Acquisition and Efficient Mechanism Design, Econometrica, 70:3, 2002, 1007-1033.

[4] F. Branco, Multiple Unite Auctions of an Indivisible Good, Economic Theory, 77-101.

[5] J. Cremer and R. P. McLean, Optimal Selling Strategies for a Discriminating Monopolist When Demands are Interdependent, Econometrica, 53, 1985, 345-361.

[6] B. Edelman, M Ostrovsky and M. Schwarz, Internet Advertising and the Generalized Second Price Auction: Selling Billions of Dollars Worth of Keywords, American Economic Review, 2007, 97:1, 242-259.8, 1996,

[7] P. Jehiel, M. Meyer-Ter-Vehn, B. Moldovanu and W. Zame, The Limits of Ex Post Implementation, Econometrica 74:3, 2006, 585-610.

[8] P. Jehiel and B. Moldovanu, Efficient Design with Interdependent Valuations, Econometrica, 69, 2001, 1237-1259.

[9] J. Levin, An Optimal Auction for Complements, Games and Economic Behavior 18, 1997, 176-192.

[10] E. Maskin, Auctions and Privatization, in Privatization, ed. by H. Siebert, Tubingen: Mohr, 115-136.

[11] E. Maskin and P. Dasgupta, Efficient Auctions, Quarterly Journal of Economics, May 2000, 341-388.

[12] R. McLean and A. Postlewaite, Implementation with Interdependent Values, 2006, University of Pennsylvania working paper.

[13] P. Milgrom and I. Segal, Envelope Theorems for Arbitrary Choice Sets, Econometrica 70, 2002, 583-601.

[14] R. Myerson, Optimal Auction Design, Mathematics of Operations Research, 6, 1981, 58-73.

[15] P. Reny and M. Perry, An Efficient Auction, Econometrica 70, 2002, 1199-1212.

[16] D. Topkis, Supermodularity and Complementarity, 1998, Princeton University Press.