Internal Implementation *

Ashton Anderson
Computer Science Department
Stanford University
ashton@cs.stanford.edu

Yoav Shoham
Computer Science Department
Stanford University

Microsoft Israel R&D Center
Herzliya Pituach, Israel
shoham@stanford.edu

Alon Altman
Computer Science Department
Stanford University
epsalon@stanford.edu

ABSTRACT

We introduce a constrained mechanism design setting called *internal implementation*, in which the mechanism designer is explicitly modeled as a player in the game of interest. This distinguished player has the opportunity to modify the game before play. Specifically, the player is able to make reliable binding commitments of outcome-specific monetary transfers to the other players in the game. We characterize the power of internal implementation for certain interesting classes of games, and show that the impact of internal implementation on the utility of the players' and the social welfare is often counterintuitive; for example, the social welfare can be arbitrarily worse after an internal implementation.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Multiagent Systems

General Terms

Economics, Theory

Keywords

Game Theory, Constrained Mechanism Design, Implementation

1. INTRODUCTION

AI, along with other areas of computer science, has embraced mechanism design as an essential tool for the design of games that incentivize agents to behave in a globally desirable way. Traditional mechanism design has imagined that the designer has unlimited freedom in designing the game. But in real multi-agent systems settings, this is often not the case [6]. For example, it may be infeasible for the designer to change the number of players or the players' strategy spaces. In recent years, a rich literature called *constrained mechanism design* has developed to address this fact, which is characterized by settings in which a designer is given a game and has limited ways of modifying it. Previous work has explored the power of different constrained mechanism designers: for example mediators and strong equilibrium are explored in [3, 5] and the power of contracts is studied in [1]. But in all of these settings, the designer is an external central planner. What happens if the designer is a player in the game of interest? Our work concentrates on this setting: we explicitly model the mechanism designer as a player in the game. It is perhaps best understood in connection with two pieces of work, one in AI and one in game theory.

The first is Monderer and Tennenholtz's k-implementation [2] setting, where an external interested party can reliably commit to outcome-based payments to the players in the game. They assume rationality only in the weakest possible sense: players do not play dominated strategies. This assumption is common to virtually all of game theory. They say an outcome is implemented if after the outcome-based commitments, it is the dominant-strategy equilibrium in the game. Clearly, given enough capital this external party can implement any outcome, so the interesting question is what is the least amount of capital k that the interested party has to pay to implement a particular outcome? Interestingly, they show that some outcomes can be implemented with no capital at all – the reliability of the interested party alone is sufficient to implement an outcome. The outcomes for which this is the case are exactly the Nash equilibria of the game.

We extend the notion of k-implementation by explicitly modeling the interested party as one of the players in the game, whom we call the implementor (hence *internal* implementation). The implementor is given the same power as Monderer and Tennenholtz's external interested party: the power to commit to arbitrary outcome-based transfers to the other players. We make the same rationality assumption, that players do not play dominated strategies.

The second, proposed and studied by Moulin [4], takes a slightly different approach: players may use action-based money-burning as a cooperative tool. That is, if a player is allowed to make binding action-based commitments to burn money, can he improve the (completely mixed) equilibria of the game for *both* players? He characterizes games which admit such "self-punishments" that result in a Pareto improvement over the equilibria in the original game.

Moulin's setting differs from ours in three important ways. First, the solution concept considered is completely mixed

*This work was supported by NSF grant IIS-0205633-001 and in part by an NSERC grant.

Cite as: Internal Implementation, Ashton Anderson, Yoav Shoham, and Alon Altman, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.

equilibrium, which lies on the opposite end of the spectrum from the dominant-strategy equilibrium concept we assume in this paper. In the same vein as Monderer and Tennenholtz, we want to make as little assumptions as possible with regard to the rationality of the players. Secondly, the offers are action-specific instead of outcome-specific, and thirdly the offers are money-burning instead of transfers. Moulin shows that under completely mixed-equilibrium these last two differences do not matter, in that his results would be the same if his assumptions matched ours. But for other solution concepts these choices make a difference.

We tackle the question of what outcomes can be optimally implemented for the benefit of the implementor. In this work, we present surprising results demonstrating that the power of implementation can bestow arbitrary gains for the implementor, while either arbitrarily decreasing or increasing social welfare and the utility of other players.

The paper is laid out as follows: in Section 2 we set our definitions and model. In Section 3 we present our main results and observations. We discuss a few of the many interesting directions for future research in Section 4 and conclude in Section 5.

2. DEFINITIONS AND MODEL

In this section we state our definitions and model. The game theory definitions mostly follow [2].

Game theory

A *game* G is a triple (N, X, U) where $N = [n]$ is the set of players, $X = X_1 \times \ldots \times X_n$, where X_i is the set of strategies available to player i, and U is a tuple $(U_1, \ldots, U_n)$, where $U_i : X \to \mathbb{R}$ is the payoff function of player i. We assume *transferable utility*, so that the payoffs of the players are represented in the same currency. Where the players and their strategies are understood, we will use $G(U)$ to denote a game in strategic form. A subscripted $-j$ refers to the set $N \backslash \{j\}$ of all players except j.

Let x_i, y_i be strategies of player i in the game $G(U)$. We say that x_i *dominates* y_i if $U_i(x_i, x_{-i}) \geq U_i(y_i, x_{-i})$ for every $x_{-i} \in X_{-i}$ and there exists some $x_{-i} \in X_{-i}$ such that a strict inequality holds. y_i is a *dominated strategy* if there exists $x_i \in X_i$ that dominates it. x_i is a *dominant strategy* if it dominates all strategies $y_i (\neq x_i) \in X_i$.

Let G be a game with payoff function vector V. $\bar{X}_i(V)$ will denote the set of non-dominated strategies for player i in the game $G(V)$, and $\bar{X}(V) = \bar{X}_1(V) \times \ldots \times \bar{X}_n(V)$. $\bar{G}(V)$ is the *non-dominated* game $(N, \bar{X}, V)$, where V is understood to mean the restriction of the payoff function to the smaller strategy space $\bar{X}$.

The *pure safety value* of i in the game $G(U)$ is the largest amount i can guarantee herself regardless of how the other players in the game play using pure strategies (also known as the pure minimax value), and is denoted $\alpha_i(G(U)) = \max_{x_i} \min_{x_{-i}} U_i(x_i, x_{-i})$. The *non-dominated pure safety value* of i in the game $G(U)$ is equal to the pure safety value of i in the non-dominated game $\bar{G}(U)$: $\bar{\alpha}_i(G(U)) = \alpha_i(\bar{G}(U))$. This is the largest amount i can guarantee herself if the players in the game avoid playing dominated strategies. Clearly $\bar{\alpha}_i(G(U)) \geq \alpha_i(G(U))$ for all i, G, and U.

Let G be a game. For each player $i \in N$, we define a *deviation matrix* D_i, where $D_i(x) = \max_{y_i \in X_i}(U_i(y_i, x_{-i}) - U_i(x_i, x_{-i}))$ for every outcome $x \in X$. $D_i(x)$ is the difference in i's utility between i's best response to x_{-i} and x. $D_i(x)$ is the amount of utility i can gain by deviating from x. If x is a Nash equilibrium, then $D_i(x) = 0$ for all i.

The *social welfare* of an outcome $x \in X$ in game G is the sum of the payoffs in that outcome: $SW_x(G) = \sum_{i \in N} U_i(x)$. The social welfare of a game G is the sum of each player's non-dominated pure safety value: $SW(G) = \sum_{i \in N} \bar{\alpha}_i(G)$.

Our setting

DEFINITION 1. *An* internal implementation I_j *is a set of offer matrices* $\{Z_i\}_{i \neq j}$ *where* $j \in N$ *is the implementor and each* Z_i *is a non-negative matrix the same size as* X *which represents the outcome-specific offers from* j *to* i.

Note that the implementor specifies a non-negative offer matrix Z_i for each player $i \neq j$, where non-negative here simply means each entry is non-negative. The special case where all Z_i's are all-zeros matrices is called the *trivial implementation*. When G is a two-player game, we will drop the subscript on the single offer matrix and refer to it simply as Z. $\mathcal{I}_j$ denotes the space of all possible implementations with implementor j.

DEFINITION 2. *The game* G' induced *by implementation* I_j *from game* G *is written* $G' = I_j(G)$, *where* $G' = (N, X, U')$, *and* U' *is specified by* $U'_i = U_i + Z_i$ *for* $i \neq j$ *and* $U'_j = U_j - \sum_{i \neq j} Z_i$.

Thus an induced game is simply the base game transformed by an implementation.

DEFINITION 3. *Let* I_j *be an implementation in game* G, *and let* $x = (x_1, \ldots, x_n) \in X$ *be a pure outcome.* x *is said to be* internally implemented in dominant strategies *by* I_j *if each* x_i *is the dominant strategy in* X_i *for all* $i \neq j$ *in the induced game* $I_j(G)$.

Note that the definition does not require the implementor j to have a dominant strategy, since j is free to select whichever action he wants. We define $\Omega_j \subset \mathcal{I}_j$ as the space of all internal implementations of an outcome in dominant strategies for player j. We also include the trivial implementation in Ω_j.

EXAMPLE 1.

G:

	C	D
C	$7, 7$	$0, 9$
D	$9, 0$	$3, 3$

In this example, let player 1 (row player) be the implementor. One possible implementation she has is to implement the outcome (C, C). Since player 2 has a profitable deviation of 2 from (C, C) (to (C, D)), she will have to promise at least 2 in (C, C). To make C a dominant strategy for player 2, she will also need to cover player 2's deviation of 3 from (D, C) to (D, D). Thus the implementation I_1 is:

Z:

	C	D
C	$2 + \epsilon$	0
D	$3 + \epsilon$	0

which transforms the game to:

$I_1(G)$:

	C	D
C	$5 - \epsilon, 9 + \epsilon$	$0, 9$
D	$6 - \epsilon, 3 + \epsilon$	$3, 3$

Note that the implementor is free to choose whichever outcome she wants in the strategy she made dominant for her opponent. Although in this example she would derive greater utility by choose (D, C) instead of (C, C), we still consider I an implementation for (C, C) (and (D, C)) because she is free to choose either. In general, an implementation I_j that implements an outcome $x = (x_j, x_{-j})$ also implements every other outcome (x_k, x_{-j}) for all k.

A major goal of this paper is to examine what a player can achieve given the ability to transform the game to one whose outcome is "clear", using minimal rationality assumptions on the players to define clear. As mentioned in Section 1, we only assume that players do not play dominated strategies. Because of this weak assumption of rationality, the result of implementations that do not implement an outcome in dominant strategies is unclear. Therefore, our main object of study in this paper will be internal implementations that implement an outcome in dominant strategies. An outcome is said to be *implemented* in the game induced by an implementation if it implements x in dominant strategies.

When x is implemented in dominant strategies I_j, the amount j actually has to pay is $k = \sum_{i \neq j} Z_i(x)$, and we refer to I_j as an *internal k-implementation* of x. We make the obvious but important remark that the implementor doesn't have to pay all of the offers, only the offers in the implemented outcome.

In k-implementation, the question is not which outcome the external agent can implement, since the external agent can implement any outcome with a sufficiently high value of k (by promising the players a large amount in the desired outcome). Instead, the value of interest for an outcome x is the smallest number k for which a k-implementation of x exists. The same is true for the implementor in internal implementation, thus for every outcome $x \in X$ we define $k(x)$ to be the smallest number k for which there exists an internal k-implementation of x in dominant strategies. Although the implementor can promise all other players a large amount in the desired outcome, the implementor cannot *profitably* internally implement all outcomes. Therefore the value of most interest to us in this setting is $\max_{x \in X} U_j(x) - k(x)$.

An important note we must make is that in this paper we focus solely on pure strategies, and pure safety levels. That is, we do not allow for mixed strategies for the players, even when those can generate a better safety level (as in Matching Pennies) or dominate a pure strategy. A simple way to extend our work to mixed strategies is to apply our results to the mixed extension of the original game, with the caveat that the implementor must be able to offer transfers based on the other players' mixed strategies and not only based on their realizations.

The set of *optimal implementations* for j in game G is

$$\mathbf{I}_j^*(G) = \operatorname*{argmax}_{I_j \in \Omega_j} \bar{\alpha}_j(I_j(G))$$

Note that the max is taken over Ω_j, for reasons discussed above. The reason we maximize $\alpha_j(I_j(G))$ instead of simply the payoff in the implemented outcome is that for games with an implemented outcome x, the implementor's non-dominated pure safety value reduces to the payoff in x, and for the few games that do not have any implementations except the trivial implementation, the non-dominated pure safety value is a conservative value that is in line with our minimal rationality assumptions.

The set of games transformed by an optimal implementation is written $\mathbf{G}_j^* = \{G^* \mid I_j^*(G) = G^*\}$, where $I_j^* \in \mathbf{I}_j^*$. Note that the set of optimal implementations is infinite, since the non-realized offers can be arbitrary without affecting the implementor's payoff.

The payoff achieved by the implementor in these optimal implementations will be denoted by

$$\beta_j(G) = \max_{I_j \in \Omega_j} \bar{\alpha}_j(I_j(G))$$

The *internal implementation value* for player j in game G is

$$IIV_j(G) = \frac{\bar{\alpha}_j(G_j^*)}{\bar{\alpha}_j(G)}$$

where $G_j^* \in \mathbf{G}_j^*$.

For a class of games $\mathbb{G}$, the internal implementation of this class of games is defined to be

$$IIV(\mathbb{G}) = \sup_{G \in \mathbb{G}, i \in N} IIV_i(G)$$

Obviously $IIV_i(G) \geq 1$, because for the trivial implementation $Z = 0$, $IIV_i(G) = 1$. Since we are maximizing over the set of non-negative offer matrices, the IIV can only be greater.

In two-player games we say that player 1 is the row player and player 2 is the column player.

3. PROPERTIES OF INTERNAL IMPLEMENTATION

As their names imply, k-implementation and internal k-implementation are closely related. The following result formally establishes that k-implementation is a strict subset of internal k-implementation.

THEOREM 1. *Let $G = (N, X, U)$ be a game, $x \in X$ be some outcome, and $j \in N$ be the implementor. Then there exists an internal k-implementation of x by j if and only if there exists a k-implementation of x in $G' = (N \backslash \{j\}, X', U')$, where $X' = X_1 \times \ldots \times X_{j-1} \times \{x_j\} \times X_{j+1} \times \ldots \times X_n$, and U' is the restriction of the payoff vector U to the smaller strategy space X'.*

PROOF. ($\Rightarrow$): Let $x \in X$ be an outcome. Assume there exists an internal k-implementation of x by j in dominant strategies. Let $\{Z_i\}_{i \neq j}$ be the offer matrices corresponding to this internal implementation. The exact same offer matrices could be used by an external agent to k-implement x in the subgame G'.

($\Leftarrow$): Assume there exists a k-implementation of x in G'. Then again, there exist offers that can be represented in offer matrices $\{Z_i\}_{i \neq j}$ for each player i in the game G'. These same offer matrices constitute an internal k-implementation of x in G. □

COROLLARY 1. *For a fixed outcome $x = (x_1, \ldots, x_n)$, implementor j can internally k-implement x with:*

$$k(x) = \sum_{i \neq j} D_i(x)$$

COROLLARY 2. *Let $x^* \in X$ be defined as follows:*

$$x^* \in \operatorname*{argmax}_{x \in X} \ (U_j(x) - k(x))$$

An implementation that internally implements x^* in dominant strategies is an optimal implementation.

PROOF. For each outcome x, the payoff j gets from internally implementing x in dominant strategies is $U_j(x) - \sum_{i \neq j} D_i(x)$. An implementation which implements the outcome that maximizes this value must be an optimal implementation.

Following the above discussion, constructing an implementation to implement x^* in dominant strategies is straightforward. Each Z_i is as follows: $Z_i(x^*) = D_i(x^*) + \epsilon$ and insurance offers $Z(x_i^*, x_{-i}) = D_i(x_i^*, x_{-i}) + \epsilon$ for all outcomes (x_i^*, x_{-i}) to make x^* dominant, and 0 elsewhere. □

Note this last result implies that finding an optimal implementation is algorithmically trivial: to find the outcome which maximizes $U_j(x) - k(x)$ we simply compute it for every outcome.

We wish to characterize exactly what powers the implementor has in our setting. What does the ability of making outcome-specific transfers give the implementor? Observe that when the implementor makes a transfer, he is both removing some of his own utility and increasing the utility of another player. In Moulin's work, there was only disposal of utility. In k-implementation, the players' utilities are only going up, since the external interested party can only increase the players' utilities. Internal k-implementation combines both.

However, removal of the implementor's utility is not a source of power. (This is in contrast with Moulin's setting, where it is the *only* source of power.) This is because it doesn't help create dominant strategies for the other players in the game, since it doesn't affect their payoffs. We are not assuming that they use iterated removal of dominated strategies, or any other stronger notion of rationality, but only that players don't play dominated strategies. Therefore, removal of the implementor's utility only hurts the implementor, and is incorporated into the model to balance the power he derives from being able to increase the utilities of the other players.

In Section 2, we explained that we restrict our attention to internal implementations of some outcome in dominant strategies. The next result shows that for two-player games, this doesn't restrict implementation power at all since an internal implementation of an outcome in dominant strategies is optimal over the whole set of implementations.

THEOREM 2. *Let G be a two-player game with implementor j. Then there exists an implementation $I_j^* \in \Omega_j$ which is optimal over $\mathcal{I}_j$:*

$$\bar{\alpha}_i(I_j^*(G)) = \max_{I_j \in \mathcal{I}_j} \bar{\alpha}_i(I_j(G))$$

Furthermore, the offer matrix Z can be completely specified by two non-negative numbers k_i and l_i, where k_i is the amount j has to actually transfer to i and l_i is the size of each non-realized payment j has to offer i.

PROOF. Without loss of generality let player 1 be the implementor. Let $I_1 \in \mathcal{I}_1$ be any implementation for player 1, and let $G' = I_1(G)$ be the game induced by I_1. $\bar{\alpha}_1(G')$ is the implementor's payoff for this implementation, and by definition this payoff is player 1's pure safety level in G'. Let $(\tilde{x}_1, \tilde{x}_2)$ be an outcome that guarantees player 1 this value: $U_1(\tilde{x}_1, \tilde{x}_2) = \bar{\alpha}_1(G')$. This outcome always exists because we are using the *pure* safety level. Notice that $U_1(\tilde{x}_1, \tilde{x}_2) \leq U_1(\tilde{x}_1, x_2')$ for all $x_2' \in X_2$, by definition of $(\tilde{x}_1, \tilde{x}_2)$. We now wish to show that player 1 can achieve $\bar{\alpha}_1(G')$ with an implementation $I_1' \in \Omega_j$ that implements an outcome in dominant strategies. There are two cases: either $\tilde{x}_2$ is a best response to $\tilde{x}_1$ or it isn't.

In the first case, $\tilde{x}_2$ is a best response to $\tilde{x}_1$. Let $I_1' = \{Z\}$, where $Z(\tilde{x}_1, \tilde{x}_2) = \epsilon$, $Z(x_1', \tilde{x}_2) = D_2(x_1', \tilde{x}_2) + \epsilon$ for all $x_1' (\neq \tilde{x}_1) \in X_1$, and 0 everywhere else. Note that all of these insurance offers can be set to the largest of them (call it l_2) without affecting player 1's payoff (since they are unrealized). By construction, $(\tilde{x}_1, \tilde{x}_2)$ is dominant in $I_1'(G)$, and as $\epsilon \to 0$, $\beta_1(I_1'(G)) \to \beta_1(I_1(G))$. Here, $k_2 = 0$ and l_2 is defined above. This covers the first case.

Now assume that $\tilde{x}_2$ is not a best response to $\tilde{x}_1$. Then there is some other outcome $\hat{x}_2$ that is a best response to $\tilde{x}_1$. Now construct I_1' exactly as before except for $(\tilde{x}_1, \hat{x}_2)$ instead of $(\tilde{x}_1, \tilde{x}_2)$: $Z(\tilde{x}_1, \hat{x}_2) = \epsilon$, insurance payments $Z(x_1', \hat{x}_2) = D_2(x_1', \hat{x}_2) + \epsilon$ for all $x_1' (\neq \tilde{x}_1) \in X_1$. Similarly to before, $(\tilde{x}_1, \hat{x}_2)$ is dominant in $I_1'(G)$, and as $\epsilon \to 0$, $\beta_1(I_1'(G)) \to U_1(\tilde{x}_1, \hat{x}_2)$. Since $U_1(\tilde{x}_1, \hat{x}_2) \geq U_1(\tilde{x}_1, \tilde{x}_2) = \beta_1(I_1(G))$, we are done. □

Thus in two-player games, the weak rationality assumption doesn't restrict the player's implementation power at all.

We will now discuss a sufficient structure of internal implementations of an outcome in dominant strategies. Let I_j be such an implementation with implementor j, and let x be the implemented outcome. For every $i \neq j$, x_i needs to be dominant in the game induced by I_j. Therefore, $Z_i(x) = D_i(x) + \epsilon$, and $Z_i(x_i, x_{-i}') = D_i(x_i, x_{-i}') + \epsilon$ for all outcomes (x_i, x_{-i}') such that $x_{-i}' \neq x_{-i}$ are sufficient. For all other outcomes y, we can take $Z_i(y) = 0$. Thus, we only have two kinds of offers: one, in x, will actually be realized. We will refer to this as k_i. The other offers, those in outcomes (x_i, x_{-i}') such that $x_{-i}' \neq x_{-i}$, are needed to ensure that x_i is dominant in the game induced by I_j. They insure player i against any possible deviation by any of the other players, and for this reason we will refer to them as *insurance offers*. Let l_i be the largest insurance offer to player i. Since insurance offers aren't realized, they can be arbitrarily high without affecting the implementor's utility. Thus we can set all insurance offers to l_i and x_i will still be dominant.

These implementations can be summarized by two vectors of numbers $\vec{k} = (k_1, \ldots, k_n)$ and $\vec{l} = (l_1, \ldots, l_n)$. $\vec{k}$ roughly represents how "far" the implemented outcome (call it x) is from being a Nash equilibrium in the original game, since $\sum_i k_i$ is the sum of profitable deviations from x that the other players have in the original game. On the other hand, l_i is the amount j has to offer in $\Pi_k |X_k| / X_i$ outcomes in order to make $x_i \in X_i$ a dominant strategy for player i. $\vec{l}$ can be thought of as the distance x is from being a dominant-strategy equilibrium after being transformed to a Nash equilibrium. Note that k_j and l_j are both 0 and are included only for simplicity.

Next we characterize the internal implementation value of certain interesting classes of games.

THEOREM 3. *1. Let $\mathcal{Z}$ be the class of two-player zero-sum games. Then*

$$IIV(\mathcal{Z}) = 1$$

2. *Let G be a game such that the highest payoffs for all players coincide in the same outcome. Then $\beta_i(G) = \max U_i$ for all $i \in N$. Thus, if we let $\mathcal{C}$ be the class of such "common-maximum" games,*

$$IIV(\mathcal{C}) = \infty$$

3. *Let $\mathcal{T}$ be the class of 2×2 games. Then*

$$IIV(\mathcal{T}) = \infty$$

PROOF. 1. Fix G to be a zero-sum game. Without loss of generality, let player 1 be the implementor. For every outcome $(x_i, x_j) \in X$, player 1's cost to internally implement (x_i, x_j) in dominant strategies is:

$$\begin{aligned} &= U_1(x_i, x_j) - \left(\max_{x_k \in X_2} \left(U_2\left(x_i, x_k\right) - U_2(x_i, x_j) \right) \right) \\ &= U_1(x_i, x_j) - \left(\min_{x_k} \left(U_1\left(x_i, x_j\right) - U_1(x_i, x_k) \right) \right) \\ &= U_1(x_i, x_j) - \left(U_1(x_i, x_j) - \min_{x_k} U_1(x_i, x_k) \right) \\ &= \min_k U_1(x_i, x_k) \end{aligned}$$

Thus player 1's payoff from an optimal implementation is:

$$\begin{aligned} \beta_1(G) &= \max_{x_1 \in X_1} \min_{x_2 \in X_2} U_1(x_1, x_2) \\ \beta_1(G) &= \alpha_1(G) \end{aligned}$$

and thus $IIV_1(G) = \bar{\alpha}_1(G)/\bar{\alpha}_1(G) = 1$ (since in zero-sum games, $\alpha_i(G) = \bar{\alpha}_i(G)$). A symmetric argument applies to player 2, and this proof naturally extends to constant-sum games.

2. In any common-maximum game the outcome with the highest payoffs for all players is clearly a Nash equilibrium. Therefore, it is internally implementable for every player for 0 cost, and all that is needed are insurance offers. Taking any common-maximum game with a parameterized maximum payoff $x \to \infty$ for all players yields the result.

3. Let G be the following Prisoner's Dilemma.

	C	D
C	x,x	0,$x+1$
D	$x+1$,0	1,1

In this game, $\bar{\alpha}_1(G) = 1$ since (D, D) is the dominant outcome, and $\beta_1(G) = x - \epsilon$ by internal implementation of (D, C). Thus $IIV_1(G) = x$, and as $x \to \infty$, $IIV_1(G) \to \infty$. Since the game is symmetric, the same argument holds for player 2. The same result naturally extends to games with larger strategy spaces and games with more players.

□

The result $IIV(\mathcal{Z}) = 1$ confirms the intuition that when the interests of two players are strictly opposed, there are no profitable internal implementations. Whatever gains are to be won from internal implementation power are lost to compensate the other player's loss. In contrast, $\beta_i(G)$ for any $G \in \mathcal{C}$ and any $i \in N$ is simply the maximum payoff in the game. In this class of "common-maximum" games, where the interests of the players are strongly (though not completely) aligned, every player achieves their maximum possible $\beta_i(G)$ no matter who implements.

The previous results are evidence of the extreme power of internal implementation. This implies that in general, in strategic situations where one player can transform the game by making outcome-specific transfers, the implementor stands to profit a lot. But the following result shows, counterintuitively, that in some games a player would prefer *another* player have internal implementation power rather than have it themselves.

THEOREM 4. *There exists a bimatrix game G with players i, j such that $\bar{\alpha}_i(I_j^*) > \bar{\alpha}_i(I_i^*)$, where $I_k^* \in \mathbf{I}_k^*$ for $k = i, j$.*

PROOF. Let G be the following game:

	L	R
U	50,100	0,0
D	101,-50	1,51

$\bar{\alpha}_1(G) = 1$ and $\bar{\alpha}_2(G) = 51$. An optimal implementation is $I_1^* = \{Z\}$ where $Z_{D,L} = 102$ and $Z = 0$ elsewhere, and the resulting payoff in the induced game $I_1^*(G)$ is $(50, 100)$. The best implementation for player 2 is the trivial implementation $I_2^* = \{\mathbf{0}\}$ where $\mathbf{0}$ is the zero matrix, and it results in the same payoff as in G. Since $100 > 51$, player 2 would benefit more from player 1's optimal implementation more than any implementation she could make. □

Since the increase in the implementor's payoff as a result of implementation can be arbitrarily high, the same is true for the increase in the social welfare as a result of implementation. However, the social welfare can also decrease, even arbitrarily low.

THEOREM 5. *Let j be the implementor. There exist G and G^*, where $G_j^* \in \mathbf{G}_j^*$, such that*

$$SW(G_j^*) - SW(G) \to -\infty$$

PROOF. Let G be the following game.

	L	R
U	3,$x-1$	0,x
D	6,-1	1,4

In G, (D, R) is the dominant-strategy equilibrium and leads to payoffs $(1, 4)$ and hence $SW(G) = 5$. But an optimal implementation for the row player is $I_1^* = \{Z\}$ where Z is the following offer matrix:

	L	R
U	$1+\epsilon$	0
D	6	0

which transforms the game to the induced game G':

	L	R
U	$2-\epsilon$,$x+\epsilon$	0,x
D	0,5	1,4

and the dominant-strategy equilibrium is now (U, L) for a payoff of $(2-\epsilon, x+\epsilon)$ and thus $SW(G') = 2+x$. Taking the limit $x \to -\infty$ yields the result. □

Now we wish to briefly consider how varying the setting studied in this paper affects the power of the implementor. There are two main components of our setting that we can vary: the first is the assumptions made on the rationality of the players, and the second is the exact specification of the ability of the implementor. We chose to assume only that players don't use dominated strategies and that the implementor can offer outcome-specific transfers, but other choices can be made. We make two interesting observations about particular settings.

The first result is in the setting where the implementor can make action-based self-punishments (utility burning). As mentioned above, if we keep our rationality assumption the same in this setting, the implementor can't implement anything because self-punishments don't affect the other players' payoffs. To make this interesting, we need to strengthen our rationality assumption. We consider the weakest possible strengthening: players assume other players don't use dominated strategies. This is tantamount to allowing the players two rounds of iterated removal of dominated strategies (first they remove everyone else's dominated strategies, then they remove their own strategies that are now dominated). An implementation in this setting will be referred to as a *self-punishing implementation*, and consists of action-based self-punishments. The following result establishes that this setting is exactly equivalent to the special case of internal 0-implementations (where the implementor doesn't make any realized payments).

THEOREM 6. *Let G be a two-player game, and let $x \in X$ be an outcome in G. Then there exists a self-punishing implementation of x in G if and only if there exists an internal 0-implementation of x in G.*

PROOF. Without loss of generality, let player 1 be the implementor.

($\Rightarrow$): Assume there is a self-punishing implementation of $x = (x_1, x_2)$ in G. Note that by definition any self-punishing implementation must result in x_2 being a dominant strategy for player 2 after the self-punishments. In particular, this means x_2 is a best response to x_1. Now consider the subgame $G' = (N', X', U')$, where $N' = \{2\}$, $X' = \{x_1\} \times X_2$, and U' is the restriction of the payoff vector U to the smaller strategy space X'. Since x_2 is a best response to x_1, x is a Nash equilibrium of this subgame. Thus there is a 0-implementation (in the k-implementation sense) of x in G'. By Theorem 1, there is also an internal 0-implementation of x in G.

($\Leftarrow$): Assume there is an internal 0-implementation of x in G. This means there are non-realized (insurance) offers that make x_2 dominant. Therefore x_2 is a best response to x_1 (if it wasn't, then there would be no way of making x_2 dominant without an offer in x). If player 1 commits to x_1 by offering large action-based self-punishments in all $x_1' (\neq x_1) \in X_1$, this would constitute a self-punishing implementation of x in G. □

The second result is that our rationality assumption (i.e. only assuming players don't play dominated strategies) is not a limiting factor in the power of internal implementation. Specifically, if we strengthen our rationality assumption to the one above (two rounds of iterated removal of dominated strategies), internal implementation is no more powerful. Let $\bar{\bar{G}}$ denote the game G after two rounds of iterated removal of dominated strategies, and let $\bar{\bar{\alpha}}_i$ be i's pure safety value in this game.

THEOREM 7. *Let G be a two-player game with implementor j, and let $I_j \in \Omega_j$ be an implementation. Then:*

$$\bar{\bar{\alpha}}_j(I_j(G)) = \bar{\alpha}_j(I_j(G))$$

where $\bar{\bar{\alpha}}_j(I_j(G))$ is j's pure safety value in $\bar{\bar{G}}'$ ($G' = I_j(G)$).

PROOF. Without loss of generality $j = 1$. Assume there is an internal k-implementation of x using the strengthened assumption of rationality. Then by definition, after implementation the opponent must have a best response to the implementor's non-dominated strategies, and $Z(x_1, x_2) = k$. A regular internal implementation with $Z(x_1, x_2) = k$, $Z(x_1', x_2) = D_2(x_1', x_2) + \epsilon$ for all $x_1' \in X_1$ and 0 everywhere else is an internal k-implementation of x. Since this holds for all x, it holds in particular for x^* (where x^* is an optimal outcome to internally implement), and the result follows. □

4. FUTURE WORK

The results presented already offer insight into the power of internal implementation, but there are a lot of potentially rich directions to pursue. Here we highlight a few such directions that we are particularly interested in.

In the current work we made very weak assumptions about the rationality of the players (we only assumed that players do not play dominated strategies). In contrast, Moulin [4] made very strong assumptions by assuming players play completely mixed equilibria. In between these two extremes lies a spectrum of assumptions that one could make about the rationality of the players. It would be interesting and useful to understand how a player's ability to transform the game to his advantage varies with the rationality assumptions of the players. Does such an ability increase monotonically with the rationality of the players? Another axis which would be interesting to explore is the particular abilities we give to the implementor. Moulin's setting has action-specific self-punishments, and in ours we have outcome-specific transfers. We think it would be beneficial to investigate how "implementation power", or its analog, varies along these two axes. Also related to this question is to address the issue of incorporating mixed strategies into our model, which we only briefly addressed.

The basic internal implementation setting presented in this paper is inherently "unfair", in that only one player has internal implementation power. What if more than one player is given internal implementation power? It seems natural to consider the case in which more than one player has the ability to change the game being played, and we believe it would be interesting to model this. Consider the following setting: given a game G, player j is given internal implementation power, and she transforms the game to G' with some implementation I_j. Then, another player k transforms this game to G'' with some implementation I_k, and so on. There are many interesting questions in this setting. Does this process converge? To what? How sensitive or robust is it to the order in which the players implement?

This setting is a form of bargaining and connections to the existing bargaining literature should be explored.

Finally, since internal implementation assumes transferable utility, it is naturally related to coalitional game theory and its related concepts such as the core. Such links should be explored in future research.

5. CONCLUSIONS

In this paper, we introduced the simple and natural internal implementation setting, which fits into the literature on constrained mechanism design but differs from it by explicitly modeling the designer as part of the game of interest. We showed that internal implementation is in general very powerful, despite making the weakest possible assumption on the rationality of the players. However, it is surprisingly not always preferable to have internal implementation power: we showed an example where one player does not have a profitable internal implementation, yet would profit greatly if the other player in the game had internal implementation power. We also showed the internal implementation value for several interesting classes of games, and examined its effects on players' utilities and social welfare.

Acknowledgement.

We thank Moshe Tennonholtz for useful conversations.

6. REFERENCES

[1] R. McGrew and Y. Shoham. Using contracts to influence the outcome of a game. In *Proceedings of the National Conference on Artifical Intelligence*, pages 238–244. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[2] D. Monderer and M. Tennenholtz. K-Implementation. *Journal of Artificial Intelligence Research*, 21:37–62, 2004.

[3] D. Monderer and M. Tennenholtz. Strong mediated equilibrium. *Artificial Intelligence*, 173(1):180 – 195, 2009.

[4] H. Moulin. Cooperation in mixed equilibrium. *Mathematics of Operations Research*, 1(3):273–286, 1976.

[5] O. Rozenfeld and M. Tennenholtz. Routing mediators. *Proceedings of the 23rd International Joint Conferences on Artificial Intelligence (IJCAI-07)*, pages 1488–1493, 2007.

[6] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press New York, NY, USA, 2008.

[7] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, 73(1-2):231 – 252, 1995. Computational Research on Interaction and Agency, Part 2.