Using Nash Implementation to Achieve Better Frugality Ratios

Chien-Chung Huang[1], Ming-Yang Kao[2], Xiang-Yang Li[3⋆] and Weizhao Wang[3]

[1] Department of Computer Science, Dartmouth College
villars@cs.dartmouth.edu
[2] Department of Electrical Engineering and Computer Science, Northwestern University
kao@cs.northwestern.edu
[3] Department of Computer Science, Illinois Institute of Technology
{xli,wangwei4}@iit.edu

Abstract. Most of the recent works on algorithmic mechanism design exploit the solution concept of dominant strategy equilibria. Such work designs a proper payment scheme so that selfish agents maximize their utility by truthfully revealing their types. It has been pointed out that these truthful mechanisms, the famous among them being the VCG mechanisms, often incur high payments and fruglity ratios. In this work, we exploit the solution concept of Nash implementation to overcome this problem. Our mechanisms induce a set of Nash equilibria so that selfish agents have incentive to act based on a Nash equilibrium. We prove that our mechanisms enjoy substantial advantages over the truthful mechanisms in terms of payment and frugality.

1 Introduction

Algorithmic mechanism design has attracted much attention of computer scientists since the seminal work of Nisan and Ronen [13]. Following their results, most of the works exploit the solution concept of dominant strategy equilibria [1, 3, 9–12]. Such work devises a proper payment scheme to ensure that, for each agent, truth-telling will maximize its utility. The VCG mechanism [4, 5, 15] is probably the most famous representative of the solutions enforcing truth-telling as the dominant strategy. We refer to this class of mechanisms as *truthful mechanisms*. A number of reasons contribute to the popularity of truthful mechanisms, including: (1) relieving each selfish agent from second-guessing whether its declared type is its best choice; and (2) maximizing the social efficiency of the outcome of the game.

In this paper, we introduce a different class of mechanisms which we call *Nash Implementation* mechanisms. Their key difference is that instead of hoping the agents to reveal their types, the proposed mechanisms induce a set of Nash equilibria so that agents maximize their profits by acting based on *any* of the induced Nash equilibria. Moreover, given any of the induced equilibria, our mechanisms guarantee that the resultant outcome would be the same as if every agent were telling the truth.

⋆ Part of the work was conducted when the author was visiting Microsoft Research Asia, Bei-Jing. The research of the author was supported in part by RGC under Grant HKBU 2104/06E and CERG under Grant PolyU-5232/07E.

It is noteworthy that, as opposed to our mechanisms, the conventional truthful mechanisms, except the dominant strategy equilibrium, may contain other Nash equilibria which may lead to undesirable outcomes. To demonstrate this possibility, consider the following auction example. Suppose that agents a and b have true types $t_a = 2$ and $t_b = 4$, respectively. The VCG mechanism works by giving the item to the highest bidder (with tie-breaking rule that favors a) and charging him the cost of the second highest bid. As is well-known, the dominant-strategy equilibrium warrants that the item is given to agent b. However, consider the following scenario. Agent a bids 10 and agent b bids 1. This is also a Nash equilibrium, but a gets the item, which is not the desired outcome.

We shall formalize the concept of Nash implementation mechanisms in Section 2. Here we first explain our motivation. Truthful mechanisms aim at soliciting the true types from the agents. Unfortunately, this is often achieved at a high cost. As has been pointed out in [2, 8], the VCG mechanism may have to pay $\Theta(n)$ times the cost of the second shortest path in the unicast game. The over-payment phenomenon of truthful mechanisms is more precisely captured by the notion of the frugality ratio [10, 14]. For instance, Karlin *et al.* [10] proved certain lower bounds of truthful mechanisms, thus implying that to acquire the true types from agents is often inherently costly.

To circumvent this over-payment problem, we relax the requirement of using the dominant-strategy equilibrium to attain the desired outcome. We allow agents to scheme together and to report their types, which correspond to a Nash equilibrium. The essence of how to Nash implementation mechanisms boils down to how to create a proper inducement so that agents profit by strategizing.

As we will show in the following sections, the most important advantages of Nash implementation mechanisms are their smaller total payments and reduced frugality ratios. Moreover, it is a more stable class of mechanisms in the sense that all Nash equilibria lead to the same desired outcome.

Our Results: To show how Nash implementation mechanisms work, we first present a polynomial-time computable mechanism $\mathcal{M}^{\mathsf{LCPA}}$ for the unicast game. We prove that its total payment is almost always smaller than that of the VCG mechanism; more generally, its payment is only slightly more than the cost of second shortest disjoint path, while the VCG mechanism might pay $\Theta(n)$ times as much. Moreover, $\mathcal{M}^{\mathsf{LCPA}}$ has a frugality ratio $2 + \epsilon$, while any truthful mechanisms has a frugality ratio at least $\Omega(\sqrt{n})$. We note that the frugality ratio of any Nash implementation for the unicast game is at least 2, therefore $\mathcal{M}^{\mathsf{LCPA}}$ is almost optimal.

Considering the more general case of binary demand games, we prove that a necessary condition for the existence of Nash implementation mechanisms is that the social choice function must be monotonic, i.e., a selected agent will still be selected if it declares a smaller cost. This condition turns out to be the same as for the truthful mechanisms. Finally, we present a general framework for designing randomized Nash implementation mechanisms for binary demand games. We prove that our framework yields a frugality ratio comparable to or significantly better than the truthful mechanisms. For example, in the vertex cover game, the frugality ratio of the VCG mechanism is $\Theta(d)$, where d is the maximum degree in the graph. Our mechanism improves this to $1 + \epsilon$, while 1 is clearly the lower bound.

Paper Structure : We review the definitions of mechanisms and introduce the necessary notation in Section 2. Seciton 3 presents a Nash implementation mechanism for the unicast game. Section 4 discusses the frugality ratio of this unicast mechanism. Section 5 gives a general framework of Nash impelmentation for binary game. Section 6 concludes. Due to space constraint, some proofs are omitted here. See full version [6] for details.

2 Preliminaries

A standard economic model for analyzing scenarios in which the agents act according to their own self-interest is as follows. There are n agents. Each agent i, for $i \in \{1, \cdots, n\}$, has some private information $\mathbf{t}_i$, called its *type*. The set of n agents define a type vector $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_n)$, which is called the *profile*. An output specification maps each type vector $\mathbf{t}$ to a set of allowed outputs. Agent i's preferences are given by a valuation function v_i that assigns a real number $v_i(\mathbf{t}_i, o)$ to each possible output o. Given the action $\mathbf{a}$ of all agents, the *utility* (often called the *profit*) of agent i is denoted as $\mathbf{u}_i(\mathbf{a}, \mathbf{t}_i)$. For a mechanism $\mathcal{M} = (\mathcal{O}, \mathcal{P})$, we assume that the utility of every agent is quasi-linear, *i.e.*, $\mathbf{u}_i(\mathbf{a}, \mathbf{t}_i) = v_i(\mathbf{t}_i, \mathcal{O}(\mathbf{a})) + \mathcal{P}_i(\mathbf{a})$.

Definition 1. A *mechanism* $\mathcal{M}$ contains two functions: an *output function* $\mathcal{O}$ and a *payment function* $\mathcal{P} = (\mathcal{P}_1, \cdots, \mathcal{P}_n)$*:*

1. For each agent i, it has a set of strategies A_i. It can choose a strategy $a_i \in A_i$.
2. For each strategy vector $a = (a_1, \cdots, a_n)$, i.e., agent i plays strategy $a_i \in A_i$, the mechanism computes an *output* $o = \mathcal{O}(a)$ and a payment $\mathcal{P}(a) = (\mathcal{P}_1(a), \mathcal{P}_2(a), \cdots, \mathcal{P}_n(a))$. The payment $\mathcal{P}_i$ is the money given to each participating agent i. If $\mathcal{P}_i < 0$, it means that the agent has to pay $-\mathcal{P}_i$ to participate in the action.

2.1 Truthful Mechanisms

By the well-known *revelation principle*, we can focus our attention on only the *direct revealing mechanisms*, in which the types are part of the strategy space A_i for each agent i. In practice, it is natural that these mechanisms should satisfy the two properties below:

- **Incentive Compatibility (IC)**: Revealing the type $\mathbf{t}_i$ is a *dominant strategy* for each agent i. In other words, for each agent i and any action a, we need that $v_i(\mathbf{t}_i, \mathcal{O}(a|^i\mathbf{t}_i)) + \mathcal{P}_i(a|^i\mathbf{t}_i) \geq v_i(\mathbf{t}_i, \mathcal{O}(a)) + \mathcal{P}_i(a)$. Here, $a|^i\mathbf{t}_i$ denotes that agent i plays strategy $\mathbf{t}_i$ and each of the other agents $j \neq i$ plays strategy a_j.
- **Individual Rationality (IR)**: This is also called *Voluntary Participation*. For each agent i and any strategy vector a, it should have $\mathbf{u}_i(a|^i\mathbf{t}_i, \mathbf{t}_i) > 0$, i.e., agent i has a non-negative profit if it reveals its true type $\mathbf{t}_i$.

Definition 2. A direct revealing mechanism is *truthful* (often referred to as *strategy-proof*) if it satisfies IR and IC.

With a truthful mechanism, the agents have no incentive to deviate from the truthful declaration because their overall utility would be no greater than it would have been if they had told the truth. Moreover, the output function guarantees that, given the declared profile $\overline{\mathbf{d}}$ and the actual type vector $\mathbf{t}$, $\mathcal{O}(\mathbf{t}) = \mathcal{O}(\overline{\mathbf{d}})$.

2.2 Nash Implementation Mechanisms

A *Nash implementation* mechanism $\mathcal{M}$ is also composed of a pair of outcome function $\mathcal{O}'$ and a payment method $\mathcal{P}$. It is associated with another social choice function $\mathcal{O}$, which maps a type vector $\mathbf{t}$ to a desirable outcome. The mechanism $\mathcal{M}$ should guarantee that its output function $\mathcal{O}'$ is "faithful" to the social choice function $\mathcal{O}$. We first define what do we mean by "faithful."

Definition 3. *The output function $\mathcal{O}$ and the social choice function $\mathcal{O}'$ have the same* range, *but may not have the same* domain, *$\mathcal{O}'(\overline{\mathbf{d}})$* equals *$\mathcal{O}(\mathbf{t})$, denoted by $\mathcal{O}'(\overline{\mathbf{d}}) \stackrel{\circ}{=} \mathcal{O}(\mathbf{t})$, if $\mathcal{O}'_i(\overline{\mathbf{d}}) = \mathcal{O}_i(\mathbf{t})$ for every agent i.*

Note that we allow the two functions $\mathcal{O}$ and $\mathcal{O}'$ to have different domains. Therefore, the declared profile $\overline{\mathbf{d}} = (\overline{\mathbf{d}}_1, \overline{\mathbf{d}}_2, \cdots, \overline{\mathbf{d}}_n)$ could be something different from a declared type vector $\mathbf{t}'$, i.e., the mechanism may require an agent to declare something other than its own type $\mathbf{t}_i$. For example, as we will show in later sections, our mechanisms demand agents to submit *two* bids to join the auction.

Having the above definition, we can now formalize what constitutes a Nash implementation mechanism.

Definition 4 (Nash Implementation Mechanism). Given a social choice function $\mathcal{O}$, a mechanism $\mathcal{M} = (\mathcal{O}', \mathcal{P})$ implements $\mathcal{O}$ in Nash equilibria if*:*

1. $\mathcal{M}$ induces a mapping $\mathbf{T} \rightarrow \overline{\mathbf{D}}$ so that a type vector $\mathbf{t} \in \mathbf{T}$ is mapped to a nonempty subset of declared profiles $\mathcal{M}(t) \subseteq \overline{\mathbf{D}}$.
2. Every declared profile $\overline{\mathbf{d}} \in \mathcal{M}(t)$ is a Nash equilibrium; conversely, every declared profile forming a Nash equilibrium is in the set $\mathcal{M}(t)$.
3. Given any declared profile $\overline{\mathbf{d}} \in \mathcal{M}(t)$, $\mathcal{O}'(\overline{\mathbf{d}}) \stackrel{\circ}{=} \mathcal{O}(\mathbf{t})$.

As mentioned earlier, we require that given any type vector, there should exist at least one Nash equilibrium for the declared profile. Moreover, every Nash equilibrium induced by this type vector should ensure that the outcome returned by $\mathcal{O}'$ is the desirable one as if everyone were behaving truthfully under the associated social choice function $\mathcal{O}$.

3 A Nash Implementation Mechanism for Unicast Using the Least Cost Path

We first review the unicast game. Assume G is a graph representing the network. Every edge e_i corresponds to a selfish agent and has a hidden cost c_i for routing. We need to "buy" a routing path from a source node s to a destination node t. This problem is solvable by the VCG mechanism [13]. Specifically, the VCG mechanism will pick up the least cost path LCP$(s, t, \overline{\mathbf{d}})$ (where $\overline{\mathbf{d}}$ will be identical with the true costs $\mathbf{c}$, as guaranteed by the mechanism), and pay each chosen edge e_i on LCP$(s, t, \overline{\mathbf{d}})$ by the amount $\mathcal{P}_k(\overline{\mathbf{d}}) = |\text{LCP}(s, t, \overline{\mathbf{d}}|^k \infty)| - |\text{LCP}(s, t, \overline{\mathbf{d}}|^k 0)|$.

As Archer and Tardos [1] pointed out, the VCG mechanism in the unicast game can be a costly solution. In certain cases, the payment is $\Theta(n)$ times the actual cost of the second shortest path from s to t. To rectify this problem, Immorlica *et al.* [8] proposed the first-price auction mechanism, in which agents are paid whatever the costs they

report if they are chosen finally. They point out that a Nash equilibrium may not exist under this mechanism, but strong ϵ-Nash equilibria always do. Moreover, the payment incurred by any strong ϵ-Nash equilibrium is never significantly more, and more often less than that by the VCG mechanism. Our new mechanism is called the *Least Cost Path Auction* (LCPA) mechanism, which is based on the work of Immorlica *et al.* [8].

The mechanism $\mathcal{M}^{\mathsf{LCPA}} = (\mathcal{O}^{\mathsf{LCPA}}, \mathcal{P}^{\mathsf{LCPA}})$ implements the function $\mathcal{O}^{\mathrm{LCP}}$ in Nash equilibria. The social choice function $\mathcal{O}^{\mathrm{LCP}}$ has bid vectors as domain and $\mathcal{O}^{\mathrm{LCP}}(\mathbf{b}) = \mathrm{LCP}(s, t, \mathbf{b})$, returning the shortest path with regard to the bid vector $\mathbf{b}$. The mechanism $\mathcal{M}^{\mathsf{LCPA}}$ requires two bids $\langle \mathbf{b}, \mathbf{b}' \rangle$ from the agents (hence the domain of $\mathcal{O}'$ is composed of two sets of bid vectors); it maps the actual cost $\mathbf{c}$ to a nonempty set of Nash equilibria; moreover, given any $\langle \mathbf{b}, \mathbf{b}' \rangle \in \mathcal{M}^{\mathsf{LCPA}}(\mathbf{c})$, $\mathcal{O}^{\mathsf{LCPA}}(\langle \mathbf{b}, \mathbf{b}' \rangle) = \mathcal{O}^{\mathrm{LCP}}(\mathbf{c})$.

The details about $\mathcal{M}^{\mathsf{LCPA}}$ are given in Algorithm 1. We explain the high-level idea here. First, we compute a shortest path $\mathrm{LCP}(s, t, \mathbf{b})$ based on the first bid $\mathbf{b}$. Then we construct another bid $\mathbf{h}$ so that given any edge i,

$$h_i = \begin{cases} b'_i \text{ if } e_i \in \mathrm{LCP}(s, t, \mathbf{b}); \\ b_i \text{ if } e_i \notin \mathrm{LCP}(s, t, \mathbf{b}). \end{cases} \tag{1}$$

In other words, for those edges which are already on the path $\mathrm{LCP}(s, t, \mathbf{b})$, they can raise their second bid in $\mathbf{b}'$ (but *only to a certain extent*, as will be explained below). Finally, we compute the shortest path $\mathrm{LCP}(s, t, \mathbf{h})$ and pay those chosen edges by the values in $\mathbf{h}$.

To adhere to Nash implementation, our primary concern is to make sure that $\mathrm{LCP}(s, t, \mathbf{h}) = \mathrm{LCP}(s, t, \mathbf{c})$. Our main trick is to seek $\mathrm{LCP}(s, t, \mathbf{c})$ by the agents' first bid $\mathbf{b}$. The difficulty lies in how to ensure that $\mathbf{b} = \mathbf{c}$. Our second concern is how to guarantee that $\mathrm{LCP}(s, t, \mathbf{h})$ returns the same path. As mentioned above, the edges on $\mathrm{LCP}(s, t, \mathbf{b})$ can alter the vector $\mathbf{h}$ by raising their second bid, and they benefit from doing so since they will be paid the amount based on their second bid if they are still chosen in $\mathrm{LCP}(s, t, \mathbf{h})$. Hence, we need to find a way to curb the over-aggressive behavior of the agents in their second bid.

To address the above two concerns, we introduce the following reward-and-punish device. In the beginning, every agent is given a small amount of premium (Line 1). They then will be asked to send a dummy packet with a certain probability (Line 2). We refer to this stage as the *broadcast stage*; this stage can be regarded as dealing out the punishment to any agent who is not giving out its true cost in its first bid. The following lemma captures the reason why we can guarantee that the first bid $\mathbf{b}$ is the same as the actual cost $\mathbf{c}$.

Lemma 1. *For each link e_i, its utility to broadcast $g_i(\mathbf{b}) = -\rho \cdot \mathbf{c}_i + f_i(s, t, \mathbf{b})$ strictly decreases in $[\mathbf{c}_i, +\infty)$ and strictly increases in $(-\infty, \mathbf{c}_i]$ on $\mathbf{b}_i$.*

The edges on $\mathrm{LCP}(s, t, \mathbf{b})$ can raise their second bid to profit. The key idea is to make sure that they can only raise their bids until a Nash equilibrium is reached. Exceeding this point, the over-bidders will not be chosen in the final path $\mathrm{LCP}(s, t, \mathbf{h})$. Moreover, they will be fined a certain amount (Line 6) because of their over aggressive behavior. Those final chosen edges are asked to provide the service and are paid (Line 5). We refer to their actual service ($s - t$ routing) as the *unicast stage*. When

Algorithm 1 Least Cost Path Auction Routing Mechanism $\mathcal{M}^{\mathsf{LCPA}} = (\mathcal{O}^{\mathsf{LCPA}}, \mathcal{P}^{\mathsf{LCPA}})$

Input: A network $G = (V, E)$, a source $s \in V$, a destination $t \in V$, $\overline{\mathbf{d}} = \langle \mathbf{b}, \mathbf{b}' \rangle$ the declared profile, and two adjustable parameters τ and γ.
Output: A mechanism $\mathcal{M}^{\mathsf{LCPA}}$.
Steps:

1: Set $\mathcal{P}_i^{\mathsf{LCPA}}(\overline{\mathbf{d}}) = f_i(s, t, \mathbf{b})$, where $f_i(s, t, \mathbf{b}) = \tau \cdot \left[b_u \cdot (n \cdot b_u - \sum_{e_j \in G - e_i} b_j) - \frac{b_i^2}{2} \right]$, for each edge $e_i \in G$, and b_u is the maximum cost any edge can declare.
2: With probability $\rho = \tau \cdot (n \cdot b_u - \sum_{e_i \in G} b_i)$, each edge is asked to send a dummy packet.
3: Compute LCP$(s, t, \mathbf{b})$; break ties by lexicographic order. For each edge e_i on LCP$(s, t, \mathbf{b})$, set $\mathbf{h}_i = \mathbf{b}'_i$, set $\mathbf{h}_i = \mathbf{b}_i$ for other edges.
4: Compute LCP$(s, t, \mathbf{h})$ and break ties according to the following rule: the edges on LCP$(s, t, \mathbf{b})$ have the highest priority while the other edges follow the lexicographic order.
5: Set $\mathcal{O}_i^{\mathsf{LCPA}}(\overline{\mathbf{d}}) = 1$ and $\mathcal{P}_i^{\mathsf{LCPA}}(\overline{\mathbf{d}}) = \mathcal{P}_i^{\mathsf{LCPA}}(\overline{\mathbf{d}}) + h_i$ for each edge on LCP$(s, t, \mathbf{h})$; *i.e.* the edges on LCP$(s, t, \mathbf{h})$ will receive the payment and relay the packet.
6: Set $\mathcal{P}_i^{\mathsf{LCPA}}(\overline{\mathbf{d}}) = \mathcal{P}_i^{\mathsf{LCPA}}(\overline{\mathbf{d}}) - \gamma \cdot |b'_i - b_i|$ for each edge in LCP$(s, t, \mathbf{b})$ but not in LCP$(s, t, \mathbf{h})$.

computing LCP$(s, t, \mathbf{b})$, we break ties by some lexicographic order (Line 3). But when we compute LCP$(s, t, \mathbf{h})$, we give priority to those edges which are already chosen in LCP$(s, t, \mathbf{b})$. This artifice guarantees the existence of a Nash equilibrium (Line 4)[4].

Nash Equilibria in $\mathcal{M}^{\mathsf{LCPA}}$ We now build a set of bid vectors and prove that they contain all the Nash equilibria induced by $\mathcal{M}^{\mathsf{LCPA}}$.

Definition 5. $\langle \mathbf{b}, \mathbf{b}' \rangle$ is said to be a *canonical form* of the bid vectors if*:*

1. $\mathbf{b} = \mathbf{c}$.
2. For each edge $e_i \in$ LCP$(s, t, \mathbf{b})$, $\mathbf{b}_i \leq \mathbf{b}'_i \leq |\text{LCP}(s, t, \mathbf{c}|^i \infty)| - |\text{LCP}(s, t, \mathbf{c}|^i 0)|$ (which is indeed the payment edge e_i gets under the VCG mechanism).
3. $\sum_{e_i \in LCP(s,t,\mathbf{b})} \mathbf{b}'_i = \sum_{e_j \in SLCP(s,t,\mathbf{c})} \mathbf{c}_j$, where SLCP$(s, t, \mathbf{c})$ is the second shortest $s - t$ path disjoint from LCP$(s, t, \mathbf{c})$.

In Lemma 2 below, we prove that all canonical bid vectors will be Nash equilibria and vice versa in Lemma 3 below. Moreover, canonical bid vectors will lead to the outcome demanded by $\mathcal{M}^{\mathsf{LCPA}}$.

Lemma 2. *(Necessity) A canonical bid vector $\langle \mathbf{c}, \mathbf{c}' \rangle$ is a Nash equilibrium for the mechanism $\mathcal{M}^{LCPA}$. Moreover, such a bid vector guarantees that the final chosen path is correct, i.e., LCP$(s, t, \mathbf{c}) =$ LCP$(s, t, \mathbf{h})$.*

Lemma 3. *(Sufficiency) Given a pair of bid vectors $\langle \mathbf{b}, \mathbf{b}' \rangle$ which forms a Nash equilibrium, then it must be canonical. Moreover, such a bid vector guarantees that the final chosen path is correct, i.e., LCP$(s, t, \mathbf{c}) =$ LCP$(s, t, \mathbf{h})$.*

Combining Lemma 2 and Lemma 3, we derive the major result of this section:

[4] The idea of using the costs of edges to break ties so as to guarantee the existence of a Nash equilibrium is mentioned by Immorlica [7, Page 66].

Theorem 1. *Mechanism $\mathcal{M}^{LCPA}$ implements the social choice function $\mathcal{O}^{LCP}$ in Nash equilibria. The social choice function $\mathcal{O}^{LCP}$ selects the shortest parth from s to t.*

By Definition 5 and Lemmas 2 and 3, the following theorem is immediate.

Theorem 2. *The total payment under the mechanism $\mathcal{M}^{LCPA}$ is at most ϵ more than that of the VCG mechanism, where ϵ can be arbitrarily small. In particular, the total payment is at most ϵ more than the actual cost of the second shortest path disjoint from the shortest path.*

Proof. The edges chosen in $\text{LCP}(s,t,\mathbf{h}) = \text{LCP}(s,t,\mathbf{c})$ will be paid at most what they would have received under the VCG mechanism, because of the second part of Definition 5. Moreover, by the third part of Definition 5, their collective payment will be at most the cost of the disjoint second shortest path. Additionally, we still have to pay an extra premium $f_i(s,t,\mathbf{b})$ to each agent i. To guarantee the total premium is smaller than ϵ, we set $\tau \leq \epsilon/(n^2 b_u^2)$. □

4 Frugality Ratio in $\mathcal{M}^{\text{LCPA}}$

In this section, we show that the frugality ratio of $\mathcal{M}^{\text{LCPA}}$ is at most $2+\epsilon$, while any Nash implementation mechanism will have frugality at least 2 in the unicast game. Hence $\mathcal{M}^{\text{LCPA}}$ has an almost optimal ratio.

We first review the definition of a frugality ratio. Consider a binary demand game $\mathscr{G} = (\mathscr{E}, \mathscr{F})$, where a set of elements $\mathscr{E}$ comprise the agents, and a certain task can be accomplished by a *feasible* team $f \in \mathscr{F}$ of elements in $\mathscr{E}$. Each element e provides a service and incurs a fixed cost $c_e \in [0,\infty)$ for performing that task. A mechanism $\mathcal{M}$ needs to find a team to perform this task and pay each element in the selected team a certain amount such that certain properties are satisfied, *e.g.*, individual rationality. In [14], Talwar proposed to measure the overpayment for a binary demand game using the *frugality*, which is defined as the total payment of the mechanism (*e.g.*, VCG) over the total cost of the *second optimal team*, which is the best team that does not intersect with the team chosen by the mechanism. The frugality notion was then generalized by Karlin *et al.* [10] to the case where the second optimal disjoint team may not exist. We review their definition here. In a binary demand game with agents $\mathscr{E}$ and feasible sets $\mathscr{F}$, let $\mathbf{T}_{opt}(\mathbf{c})$ be the feasible team with the optimal cost and $\upsilon(\mathbf{c})$ be the solution of the following problem.

$$\upsilon(\mathbf{c}) = \min \sum_{e_i \in \mathbf{T}_{opt}(\mathbf{c})} x_i \quad subject\ to \tag{2}$$

1. $x_i \geq c_i$ for every agent $e_i \in \mathscr{E}$;
2. $\sum_{e_i \in \mathbf{T}_{opt}(\mathbf{c})-F} x_i \leq \sum_{e_j \in F - \mathbf{T}_{opt}(\mathbf{c})} c_j, \quad \forall F \in \mathscr{F}$; and
3. for every $e_i \in \mathbf{T}_{opt}(\mathbf{c})$, there is a team $F \in \mathscr{F}$ such that $e_i \notin F$ and $\sum_{e_i \in \mathbf{T}_{opt}(\mathbf{c})-F} x_i = \sum_{e_j \in F - \mathbf{T}_{opt}(\mathbf{c})} c_j$.

Definition 6. *The frugality, denoted by $\phi_{\mathcal{M}}$, of a truthful mechanism $\mathcal{M}$ for a given game $\mathscr{G}$ is defined as $\phi_{\mathcal{M}} = \sup_{\mathbf{c}} \frac{\mathcal{P}(\mathbf{c})}{\upsilon(\mathbf{c})}$, i.e., the maximum possible ratio of the total payment by mechanism $\mathcal{M}$ over $\upsilon(\mathbf{c})$. The frugality of a game $\mathscr{G} = (\mathscr{E}, \mathscr{F})$ is defined as $\phi_{(\mathscr{E},\mathscr{F})} = \inf_{\mathcal{M}} \phi_{\mathcal{M}}$, where the infimum is taken over all truthful mechanisms $\mathcal{M}$.*

We can extend the frugality definition for truthful mechanisms to Nash implementation truthful mechanisms in a natural way as follows.

Definition 7. *The frugality, denoted by $\phi_{\mathcal{M}}$, of a Nash implementation mechanism $\mathcal{M}$ for a given game $\mathscr{G}$ is defined as $\phi_{\mathcal{M}} = \sup_{\mathbf{c}} \frac{\mathcal{P}(\overline{\mathbf{d}})}{v(\mathbf{c})}$, i.e., the maximum possible ratio of the total payment of the mechanism $\mathcal{M}$ over $v(\mathbf{c})$. Here $\overline{\mathbf{d}}$ is a Nash equilibrium under $\mathcal{M}$ when the true cost vector of agents is $\mathbf{c}$. The frugality of a game $\mathscr{G} = (\mathscr{E}, \mathscr{F})$ based on output method $\mathcal{O}$ is defined as $\phi_{(\mathscr{E},\mathscr{F},\mathcal{O})} = \inf_{\mathcal{M}} \phi_{\mathcal{M}}$, where the infimum is taken over all* Nash implementation truthful *mechanisms $\mathcal{M}$ with respect to method $\mathcal{O}$. The frugality of a game $\mathscr{G} = (\mathscr{E}, \mathscr{F})$ is defined as $\phi_{(\mathscr{E},\mathscr{F})} = \inf_{\mathcal{M}} \phi_{\mathcal{M}}$, where the infimum is taken over all Nash implementation mechanisms $\mathcal{M}$.*

We show below that $\mathcal{M}^{\mathsf{LCPA}}$ achieves frugality $2+\epsilon$, and this ratio is almost optimal. Before we proceed, we should point out that Karlin *et al.* [10] proved that the VCG mechanism has frugality $\Theta(n)$ and *any* truthful mechanism for the unicast game has frugality $\Omega(\sqrt{n})$. Clearly, Nash implementation mechanisms have better frugality.

Theorem 3. *The frugality of the mechanism $\mathcal{M}^{\mathsf{LCPA}}$ is $2 + \epsilon$, and the frugality of any Nash implementation mechanism based on LCP is at least 2.*

5 Nash Implementation Mechanisms for Binary Demand Games

In this section, we give a general framework for a Nash implementation mechanism. Assuming a social choice function $\mathcal{O}^{opt}$ which returns the team with minimum cost, Algorithm 2 gives a mechanism implements $\mathcal{O}^{opt}$ in Nash equilibria. Without loss of generality, we assume that the given set system $(\mathscr{E}, \mathscr{F})$ is *upwards closed*, i.e. for every $S \in \mathscr{F}$ and every superset T with $S \subseteq T \subseteq \mathscr{E}$, we have $T \in \mathscr{F}$. To avoid triviality, we assume that the system is *monopoly-free*, i.e., there is no element present in all feasible teams. For any set $T \subseteq \mathscr{E}$, let $w(T, \mathbf{c}) = \sum_{e \in T} c_e$ be the weight of the team T under cost vector $\mathbf{c}$.

The intuition for Algorithm 2 is similar to the Algorithm 1. In the beginning, every agent is given a premium (Line 1). There is a chance that it will be recruited into the team (Line 2) even if it does not belong to the optimal team. We make the final decision based on $\mathbf{h}$ and punish the over-greedy bidders (Line 6).

Theorem 4. *With probability $1 - \epsilon$, for arbitrarily small ϵ, the mechanism $\mathcal{M}^{out}$ implements $\mathcal{O}^{opt}$ in Nash equilibria.*

The proof can be obtained through Lemmas 4 and 5 below. Here we only mention that there is a small probability that the final recruited team is a superset of the optimal team. To make sure the probability of this happening is smaller than ϵ, we make $\tau \leq \epsilon/(n^2 b_u)$.

Similar to the mechanism $\mathcal{M}^{\mathsf{LCPA}}$ for unicast game, any Nash equilibrium for the mechanism $\mathcal{M}^{out}$ has the following properties.

Lemma 4. *If $\overline{\mathbf{d}} = \langle \mathbf{b}, \mathbf{b}' \rangle$ is a Nash equilibrium, then (1) $\mathbf{b} = \mathbf{c}$; (2) $\mathbf{b}'_i = \mathbf{c}_i$ if $i \notin T_{opt}(\mathbf{c})$ and $\mathbf{b}'_i \geq \mathbf{c}_i$ otherwise; (3) for any feasible team T, $w(T, \mathbf{b}') \geq w(T_{opt}(\mathbf{c}), \mathbf{b}')$.*

Algorithm 2 General Framework to Design Nash Implementation Mechanisms for Binary Demand Game

Input: A set system $(\mathscr{E}, \mathscr{F})$, $\overline{\mathbf{d}} = \langle \mathbf{b}, \mathbf{b}' \rangle$ the declared profile, and two adjustable parameters τ and γ.
Output: A mechanism implementing $\mathcal{M}^{out} in Nash equilibria$.
Steps:

1: Set $\mathcal{P}_i^{out}(\overline{\mathbf{d}}) = f_i(s,t,\mathbf{b})$, where $f_i(\mathbf{b}) = \tau \cdot \left[b_u \cdot (n \cdot b_u - \sum_{e_j \in \mathscr{E} - e_i} b_j) - \frac{b_i^2}{2} \right]$ for each agent i.
2: With probability $\rho = \tau \cdot (n \cdot b_u - \sum\limits_{e_i \in \mathscr{E}} b_i)$, we select all elements and ask them to perform the service.
3: Find the optimal teams and break ties by favoring teams with bigger sizes. After that, break ties by lexicographic order. For every agent i on $T_{opt}(\mathbf{b})$, set $\mathbf{h}_i = \mathbf{b}'_i$; otherwise set $\mathbf{h}_i = \mathbf{b}_i$.
4: Find the optimal teams on $T_{opt}(\mathbf{h})$ and break ties according to the rule that teams containing members in $T_{opt}(\mathbf{b})$ have the highest priority and if two teams have the same cost, choose the one that contains more agents in $T_{opt}(\mathbf{b})$.
5: Set $\mathcal{O}_i^{out}(\overline{\mathbf{d}}) = 1$ and $\mathcal{P}_i^{out}(\overline{\mathbf{d}}) = \mathcal{P}_i^{out}(\overline{\mathbf{d}}) + \mathbf{h}_i$ for each agent in $T_{opt}(\mathbf{h})$.
6: Set $\mathcal{P}_i^{out}(\overline{\mathbf{d}}) = \mathcal{P}_i^{out}(\overline{\mathbf{d}}) - \gamma \cdot |b'_i - b_i|$ for each agent on $T_{opt}(\mathbf{b}) - T_{opt}(\mathbf{h})$.

Before we present our main results in Theorem 5 below, we first introduce the notion of the worst and best Nash equilibria for binary demand games. Consider the linear program (2); interestingly, each solution that satisfies all constraints corresponds to a subvector of $\mathbf{b}'$ in a Nash equilibrium $\overline{\mathbf{d}}$.

Lemma 5. *There exists an one-to-one mapping between the feasible solution satisfying the constraints of linear program (2) and* $\mathbf{b}'$ *of a Nash equilibrium* $\overline{\mathbf{d}}$ *for the mechanism* $\mathcal{M}^{out}$.

Recall that the main part of payment is the sum of the bids in the subvector. Thus, for convenience of presentation, we call each solution that meets all constraints of the linear program (2) a *NE bid*. Thus, the solution to linear program (2) is the *minimum NE bid*, denoted by $\mathbf{X}^{\min}$. Next, we introduce a counterpart of the linear program (2). Let $\varrho(\mathbf{c})$ be the solution of the following linear program:

$$\varrho(\mathbf{c}) = \max \sum_{e_i \in T_{opt}} x_i \quad subject\ to \tag{3}$$

1. $x_i \geq c_i$ for each agent $e_i \in \mathscr{E}$;
2. $\sum_{e_i \in T_{opt}(\mathbf{c}) - T} x_i \leq \sum_{e_j \in T - T_{opt}(\mathbf{c})} c_j, \quad \forall T \in \mathscr{F}$; and
3. For every $e_i \in T_{opt}(\mathbf{c})$, there is a $T \in \mathscr{F}$ such that $e_i \notin T$ and $\sum_{e_i \in T_{opt}(\mathbf{c}) - T} x_i = \sum_{e_j \in T - T_{opt}(\mathbf{c})} c_j$.

The solution to linear program (3) is *maximum NE bids*, denoted by $\mathbf{X}^{\max}$. The ratio of the maximum and minimum NE bids is $\lambda = \frac{\sum_{e_i \in T_{opt}(\mathbf{c})} x_i^{\max}}{\sum_{e_i \in T_{opt}(\mathbf{c})} x_i^{\min}}$, which is called *NE ratio*. Interestingly, NE ratio always equals the ratio between the *price of stability* over the *price of the anarchy*. Let $OPT(\mathbf{c})$ be the globally optimum solution. Recall

that the price of stability is defined as $\frac{\varrho(\mathbf{c})}{OPT(\mathbf{c})}$, i.e., the ratio of the value of the largest Nash equilibrium over the optimum solution. The price of anarchy is defined as $\frac{\upsilon(\mathbf{c})}{OPT(\mathbf{c})}$, i.e., the ratio of the value of the smallest Nash equilibrium over the optimum solution. Next, we show a relationship between the Nash equilibrium and the frugality of a Nash implementation mechanisms.

Theorem 5. *Let $\mathcal{M}^{out} = (\mathcal{O}', \mathcal{P})$ be the mechanism computed by Algorithm 2. Then its frugality is $\lambda + \epsilon$, where λ is the NE ratio and ϵ is a positive number depending on parameters τ and γ.*

Proof. From Lemma 5, if $\overline{\mathbf{d}}$ is a Nash equilibrium of $\mathcal{M}^{out}$, then $\mathbf{b}'$ satisfies all the constraints of linear program (3). Thus, $\sum_{e_i \in T_{opt}(\mathbf{c})} b'_i \leq \sum_{e_i \in T_{opt}(\mathbf{c})} x_i^{\max}$. Recall that the total payment is

$$\begin{aligned}\mathbb{P}(\overline{\mathbf{d}}) &= \sum_{e_i} f_i(\mathbf{b}) + \sum_{e_i \in T_{opt}(\mathbf{c})} b'_i \leq \sum_{e_i \in T_{opt}(\mathbf{c})} x_i^{\max} + \tau \cdot n^2 b_u^2 \\ &= (1+\epsilon) \cdot \sum_{e_i \in T_{opt}(\mathbf{c})} x_i^{\max} = (1+\epsilon) \cdot \lambda \cdot \sum_{e_i \in T_{opt}(\mathbf{c})} x_i^{\min}.\end{aligned}$$

This proves the theorem. □

Recall that in order to compute the NE ratio λ, we need to solve linear program (2), which is still an open problem even for some special binary demand games, *e.g.*, the shortest path game. However, we are able to get tight bounds of the NE ratio for some binary demand games using different approaches. Before we show how to bound the NE ratio, we define some terms first.

We call a feasible team T a *base-team* if removing any elements from T makes it unfeasible. Given a team $T_1 \subseteq T$ where T is a base team, we say that team T_2 *covers* T_1 through T if $(T - T_1) \bigcup T_2$ is also a feasible team. T_2 covers an element e_i through T if there exists an $T_1 \subseteq T$ such that $e_i \in T_1$ and T_2 covers T_1 through T.

A team set $\mathscr{T}$ is a *team-cover* of a base-team T if for each element $e_i \in T$, there exists a team $T_i \in \mathscr{T}$ such that T_i covers e_i through T. A team cover $\mathscr{T}$ of T is a *minimal team cover* (MTC) of T if (1) $\mathscr{T} - T_i$ is not a team cover of T for any $T_i \in \mathscr{T}$; and (2) for any team $T_i \in \mathscr{T}$ and $e_j \in T$, $(\mathscr{T} - T_i) \bigcup (T_i - e_j)$ is not a team cover. Given a base team T and its minimal team cover $\mathscr{T}$, the degree of an element $e_i \in T$ is the number of different teams in $\mathscr{T}$ that covers e_i through T, denoted by $\deg_i(T, \mathscr{T})$. The maximum degree of T and $\mathscr{T}$ is $\deg^{\max}(T, \mathscr{T}) = \max_{e_i \in T} \deg_i(T, \mathscr{T})$; the minimum degree of T and $\mathscr{T}$ is $\deg^{\min}(T, \mathscr{T}) = \min_{e_i \in T} \deg_i(T, \mathscr{T})$. The *degree ratio* of T and $\mathscr{T}$ is $\mathrm{dr}(T, \mathscr{T}) = \frac{\deg^{\max}(T, \mathscr{T})}{\deg^{\min}(T, \mathscr{T})}$ and degree ratio of game $\mathscr{G}$ is $\mathrm{dr}(\mathscr{G}) = \max_{T, \mathscr{T}} \mathrm{dr}(T, \mathscr{T})$.

Theorem 6. *Assume we are given a fixed cost vector $\mathbf{c}$. We have $\lambda \leq dr(\mathbf{c})$. Here $dr(\mathbf{c}) = \max_{\mathscr{T}} dr(T_{opt}(\mathbf{c}), \mathscr{T})$ where $\mathscr{T}$ is a minimal team-cover of $T_{opt}(\mathbf{c})$.*

Proof. From the definition of $\mathbf{b}^{\min}$, for every $e_i \in T_{opt}(\mathbf{c})$, there exists a $T_i \in \mathscr{F}$ such that $e_i \notin T_i$ and
$\sum_{e_j \in T_{opt}(\mathbf{c}) - T_i} x_j^{\min} = \sum_{e_j \in T_i - T_{opt}(\mathbf{c})} c_j$. Let $T'_i = T_i - T_{opt}(\mathbf{c})$ and $F_i = T_{opt}(\mathbf{c}) -$

T_i. Then $\mathscr{T} = \{T'_i : e_i \in T_{opt}(\mathbf{c})\}$ is a team-cover of $T_{opt}(\mathbf{c})$. Let $\mathscr{T}^\star$ be any minimal team-cover that is a subset of $\mathscr{T}$. Without loss of generality, we assume $\mathscr{T}^\star = \{T'_1, \ldots, T'_k\}$. Thus, $\mathrm{dr}(\mathbf{c}) \cdot \sum_{e_j \in T_{opt}(\mathbf{c})} x_j^{\min} \geq \sum_{i=1}^{k} \sum_{e_j \in F_i} x_j^{\min} = \sum_{i=1}^{k} \sum_{e_j \in T'_i} c_j$. On the other hand, T'_i covers F_i through T, $\sum_{e_j \in F_i} x_j^{\max} \leq \sum_{e_j \in T'_i} c_j$. Thus,

$$\sum_{e_j \in T_{opt}(\mathbf{c})} x_j^{\max} \leq \sum_{i=1}^{k} \sum_{e_j \in F_i} x_j^{\max} \leq \sum_{i=1}^{k} \sum_{e_j \in T'_i} c_j \leq \mathrm{dr}(\mathbf{c}) \cdot \sum_{e_j \in T_{opt}(\mathbf{c})} x_j^{\min}.$$

Therefore, $\lambda \leq \mathrm{dr}(\mathbf{c})$, which proves the theorem. □

frugality ratio	VCG mechanism	Minimum frugality Truthful mechanism	Nash Implementation mechanism $\mathcal{M}^{out}$
Matroid Game	1	1	$1+\epsilon$
Unicast Game	$\Theta(n)$	$\Omega(\sqrt{n})$	$2+\epsilon$
Vertex Cover Game	$\Theta(d)$	$\Omega(\sqrt{d})$	$1+\epsilon$
Edge Cover Game	$\Theta(n)$	$\Omega(n)$	$d-1+\epsilon$

Table 1. Summary of the frugalities for some binary demand games. Here, d is the maximum degree of a vertex in the given graph.

Next we show how to find the NE ratio via an example of the vertex cover game. In the vertex cover game, given a graph $G = (V, E)$ and each vertex v_i has a cost c_i, we need to find a subset $S \in V$ such that each edge has at least one vertex in S. Here every vertex is an element and a base-team is a minimum vertex cover. For the vertex cover game, we have the following lemma regarding $\mathrm{dr}(\mathbf{c})$.

Lemma 6. *For the vertex cover game, given a fixed cost vector* $\mathbf{c}$*, we have dr*$(\mathbf{c}) = 1$.

By definition, the NE ratio λ is at least 1. From Theorem 6 and Lemma 6, $\lambda \leq 1$. Thus, the NE ratio of a vertex set cover game is exactly one.

Theorem 7. *For the vertex cover game, the frugality of the Nash implementation mechanism computed by Algorithm 2 is* $1+\epsilon$ *and the frugality of the VCG mechanism is* $n-1$*, where* n *is the number of the vertices. For any truthful mechanism, the frugality is at least* $\Omega(\sqrt{n})$.

The frugality ratio of the edge cover game under various mechanisms is summarized in Theorem 8 below.

Theorem 8. *Given a graph* G *with maximum degree* d*, the frugality ratio of the Nash implementation mechanism computed by Algorithm 2 is* $d-1+\epsilon$ *and the frugality ratio of the VCG mechanism is* $n-1$*. Any truthful mechanism has the frugality ratio at least* $\Omega(n)$

Table 1 summarizes the frugality ratios of several binary demand games under three different mechanisms: the VCG mechanism, the minimum frugality ratio truthful mechanism and our Nash implementation mechanism $\mathcal{M}^{out}$ defined by Algorithm 2. We point out that sometimes the frugality ratio does not fully capture the over-payment of a mechanism. For example, even though the VCG mechanism has a comparable frugality ratio to Algorithm 2, the actual payment of the former can be $\Theta(n)$ times the latter. For example, consider the following spanning tree game. Given a cycle with only one edge having cost 1 while the others having cost 0. The VCG mechanisms pays $n-1$ while our mechanism only $1+\epsilon$.

6 Discussions

In this paper, we propose a class of mechanisms which implement social choice functions in Nash equilibria. Instead of relying on dominant strategy equilibria, these mechanisms aim at ensuring that any attainable Nash equilibrium will lead to the desirable outcome. We show that these mechanisms enjoy advantages over truthful mechanisms in terms of reduced payments and better frugality ratios.

Acknowledgment

We thank Professor Xiaotie Deng and his students for some helpful discussions.

References

1. A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, pages 482–491, Las Vegas, Nevada, 2001. IEEE Computer Society.
2. A. Archer and E. Tardos. Frugal path mechanisms. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 991–999, Philadelphia, PA, USA, 2002. Society for Industrial and Applied Mathematics.
3. V. Auletta, R. Prisco, P. Penna, and P. Persiano. Deterministic truthful approximation schemes for scheduling related machines. In *Proceedings of the 21st Symposium on Theoretical Aspects of Computer Science*, pages 608–619, Le Corum, Montpellier, France, March 2004.
4. E. H. Clarke. Multipart pricing of public goods. *Public Choice*, pages 17–33, 1971.
5. T. Groves. Incentives in teams. *Econometrica*, pages 617–631, 1973.
6. C.-C. Huang, M.-Y. Kao, X.-Y. Li, and W. Wang. Using nash implementation to achieve better frugality ratios note = Technical Report TR2007-604, Computer Science Department, Dartmouth College, year =.
7. N. Immorlica. *Computing With Strategic Agents*. PhD thesis, MIT, 2005.
8. N. Immorlica, D. Karger, E. Nikolova, and R. Sami. First-price path auctions. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 203–212, 2005.
9. M.-Y. Kao, X.-Y. Li, and W. Wang. Towards truthful mechanisms for binary demand games: A general framework. In *Proceedings of the 6th ACM conference on Electronic Commerce*, pages 213–222, 2005.
10. A. Karlin, D. Kempe, and T. Tamir. Beyond VCG: Frugality of truthful mechanisms. In *46th Annual IEEE Symposium on Foundations of Computer Science*, pages 615–626, 2005.
11. D. Lehmann, L. I. Ocallaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of ACM*, 49(5):577–602, 2002.
12. A. Mu'alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions: extended abstract. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 379–384, Edmonton, Alberta, Canada, 2002. American Association for Artificial Intelligence.
13. N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proceedings of the 31st Annual Symposium on Theory of Computing*, pages 129–140, Atlanta, Georgia, United States, 1999.
14. K. Talwar. The price of truth: Frugality in truthful mechanisms. In *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, pages 608–619, London, UK, 2003. Springer-Verlag.
15. W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, pages 8–37, 1961.