An Asymptotically Optimal VCG Redistribution Mechanism for the Public Project Problem

Mingyu Guo
School of Computer Science, University of Adelaide, Australia
mingyu.guo@adelaide.edu.au

Abstract

We study the classic public project problem, where a group of agents need to decide whether or not to build a non-excludable public project. We focus on efficient, strategy-proof, and weakly budget-balanced mechanisms (VCG redistribution mechanisms). Our aim is to maximize the worst-case efficiency ratio — the worst-case ratio between the achieved total utility and the first-best maximum total utility. Previous studies have identified the optimal mechanism for 3 agents. It was also conjectured that the worst-case efficiency ratio approaches 1 asymptotically as the number of agents approaches infinity. Unfortunately, no optimal mechanisms have been identified for cases with more than 3 agents. We propose an asymptotically optimal mechanism, which achieves a worst-case efficiency ratio of 1, under a minor technical assumption: we assume the agents' valuations are rational numbers with bounded denominators. We also show that if the agents' valuations are drawn from identical and independent distributions, our mechanism's efficiency ratio equals 1 with probability approaching 1 asymptotically. Our results significantly improve on previous results. The best previously known asymptotic worst-case efficiency ratio is 0.102. For non-asymptotic cases, our mechanisms also achieve better ratios than all previous results.

1 Introduction

1.1 Problem Description

The public project problem is a classic mechanism design problem that has been studied extensively in economics and computer science [Mas-Colell *et al.*, 1995; Moore, 2006; Moulin, 1988]. In this problem, n agents decide whether or not to build a non-excludable public project (*e.g.*, a bridge that can be accessed by everyone). There are two possible outcomes: *build* and *not build*. Without loss of generality, we assume that the project costs 1. We use θ_i $(0 \leq \theta_i \leq 1)$[1] to denote agent i's valuation for the project if it is built. If the decision is not to build, then everyone retains her share of the cost, which equals $1/n$. That is, agent i's valuation is either θ_i (if the project is built) or $1/n$ (if the project is not built). Our model is identical to those used in existing literature on VCG redistribution mechanisms for public project problems [Guo *et al.*, 2011; Naroditskiy *et al.*, 2012; Guo *et al.*, 2013; Guo, 2016; Guo and Shen, 2017].

We focus on mechanisms that are *efficient*, *strategy-proof*, and *weakly budget-balanced* (the mechanisms should never incur deficits). Due to a characterization result from [Holmström, 1979], the only mechanisms with the above properties are the *VCG redistribution mechanisms*. Under a VCG redistribution mechanism, we first run the VCG mechanism, then every agent receives (or pays additionally) an amount that is independent of her own type as her redistribution payment. Since an agent's redistribution is independent of her own type, this guarantees that the incentives are not affected, therefore maintaining efficiency and strategy-proofness of VCG. To maintain weakly budget-balance, the total amount redistributed should never exceed the total VCG payment. Essentially, VCG redistribution mechanisms are just Groves mechanisms that are weakly budget-balanced.

We aim to maximize the *worst-case efficiency ratio*:

Definition 1. *(Due to Moulin [2009]) A mechanism's worst-case efficiency ratio is the worst-case ratio between the achieved total utility and the* first-best *total utility. Here, the worst case is taken over all possible type profiles.*

The first-best total utility is defined as the maximum total utility assuming that the agents act unselfishly.

One way to interpret the first-best total utility is that we assume the agents would voluntarily reveal their private types. We then solve the mechanism design problem as a pure optimization problem, ignoring incentives altogether. The first-best total utility often cannot be achieved by strategy-proof mechanisms. The worst-case ratio between the achieved total utility and the first-best total utility is then an indication of the cost of implementing truthfulness.

[1] Naroditskiy *et al.* [2012] showed that for the objective of maximizing worst-case efficiency ratios, it is without loss of generality to assume that the agents' valuations (the θ_i) are bounded above by the project cost (*i.e.*, 1). Any mechanism can be easily generalized to handle θ_i values that are larger than 1, and the generalization does not reduce the mechanism's worst-case efficiency ratio.

Under our public project problem, if the agents act unselfishly, then the optimal total utility (considering the weakly budget-balance constraint) is achieved by the following trivial mechanism: no payments are needed, build if and only if the agents' total valuation exceeds the project cost. It should be noted that this trivial mechanism is only used as a benchmark as it is not strategy-proof. The first-best total utility equals $\max\{\sum \theta_i, 1\}$.

It should be noted that the worst-case efficiency ratio is at most 1 for weakly budget-balanced mechanisms. Under a minor technical assumption, we are able to derive a mechanism whose worst-case efficiency ratio approaches 1 as the number of agents approaches infinity. Therefore, even though we focus on strategy-proof and efficient mechanisms, for asymptotic cases, strategy-proofness and efficiency are "free constraints" — even if we drop these constraints, we cannot get better worst-case efficiency ratios. In other words, we have identified an asymptotically optimal weakly budget-balanced mechanism for the public project problem, and the resulting optimal mechanism is efficient and strategy-proof.

1.2 Related Research and Our Contributions

VCG redistribution mechanisms have been studied extensively in resource allocation settings. For the single-item auction, Cavallo [Cavallo, 2006] identified an asymptotically optimal VCG redistribution mechanism, whose worst-case efficiency ratio approaches 1 as the number of agents approaches infinity. Guo [Guo, 2011] proposed a similar asymptotically optimal mechanism for resource allocation with gross substitutes (*i.e.*, multi-unit auctions with non-increasing marginal values, heterogeneous-item auctions with single demand). For non-asymptotic cases with specific numbers of agents, optimal VCG redistribution mechanisms have also been identified for various resource allocation settings, such as multi-unit auctions and heterogeneous-item auctions [Guo and Conitzer, 2009; Moulin, 2009; Gujar and Narahari, 2011; Guo, 2012]. Manisha *et.al.* [Manisha *et al.*, 2018] studied applying Deep Learning to the design of redistribution mechanisms.

VCG redistribution mechanisms have also been studied extensively for the public project problem [Guo *et al.*, 2013; Guo *et al.*, 2011; Naroditskiy *et al.*, 2012; Guo, 2016; Guo and Shen, 2017]. Unfortunately, previous research on the public project problem has been less successful. Our paper significantly improves on previous results on multiple aspects. We summarize the related results from previous research and our contributions as follows:

Previous Results (PR) and Our Contributions (OC)

PR Naroditskiy *et al.* [2012] conjectured that for the public project problem, the asymptotic worst-case efficiency ratio can reach 1. The authors discretized the type spaces and used Automated Mechanism Design to derive the upper bounds on the worst-case efficiency ratios for small n values. The upper bounds follow a pattern, which shows that as n approaches infinity, the upper bound approaches 1. This is a conjectured result because the upper bounds are not shown to be strict. Also, the pattern on the upper bounds was based on experimental results on small n values, so it may or may not hold for larger n.

OC We confirm that the asymptotic worst-case efficiency ratio can indeed reach 1, under a minor technical assumption: we assume the agents' valuations are rational numbers with bounded denominators. For example, if the agents' valuations are expressed in terms of real-life currency values, then our assumption holds.

PR Guo [2016] proposed a mechanism with a constant asymptotic worst-case efficiency ratio of 0.102.

OC With the aforementioned technical assumption, we have achieved a ratio of 1. Even without the assumption, we have identified another mechanism with a constant asymptotic worst-case efficiency ratio of 0.5.

PR For non-asymptotic cases with specific numbers of agents, several mechanisms have been proposed that achieved constant worst-case efficiency ratios, including one heuristic-based **SBR** mechanism [Naroditskiy *et al.*, 2012], another heuristic-based **ABR** mechanism [Guo, 2016], and mechanisms derived via Automated Mechanism Design (**AMD**) [Guo and Shen, 2017]. These mechanisms are either computationally expensive to generate [Guo and Shen, 2017] or the worst-case ratios have to be calculated via numerical methods [Naroditskiy *et al.*, 2012; Guo, 2016]. As a result, these mechanisms' ratios have only been calculated for at most 10 agents.

OC We propose a mechanism with better worst-case efficiency ratios than all previous mechanisms. Furthermore, the mechanism's implementation takes $O(n^3)$ complexity. The worst-case efficiency ratio has been characterized analytically ($\frac{n+1}{2n}$). That is, unlike previous results, our mechanism can also be used for large n values. We present the competitive ratios of different mechanisms in the following table. The column **OC** contains the efficiency ratios of our mechanisms. The column **UB** contains the *conjectured* upper bounds on the efficiency ratios [Naroditskiy *et al.*, 2012].

n	SBR	ABR	AMD	OC	UB
3	0.333	0.334	0.667	0.667	0.667
4	0.354	0.459	0.600	0.625	0.666
5	0.360	0.402	0.545	0.600	0.714
6	0.394	0.386	0.497	0.583	0.868
7	n too large	0.360	0.465	0.571	0.748
8	n too large	0.352	0.444	0.563	0.755
9	n too large	0.339	0.422	0.556	0.772
10	n too large	0.336	0.405	0.550	0.882

OC Lastly, we also proved that if the agents' valuations are drawn from identical and independent distributions, our mechanisms' asymptotic efficiency ratio equals 1 with probability close to 1 (approaching 1 asymptotically). For example, if every agent's valuation is drawn *i.i.d.* from the uniform $U(0, 1)$ distribution, then with 20 agents, the efficiency ratio is 1 with probability at least 0.954. With 50 agents, the probability becomes at least 0.991.

2 Formal Model Description

For the public project problem, anonymous[2] VCG redistribution mechanisms have the following form [Naroditskiy *et al.*, 2012]:

- Build the public project if and only if $\sum \theta_i \geq 1$.
- If the decision is to build, then agent i receives $\sum_{j \neq i} \theta_j - h(\theta_{-i})$.
- If the decision is not to build, then agent i receives $(n-1)/n - h(\theta_{-i})$.
- h is an arbitrary function and θ_{-i} refers to the types from the agents other than i herself.

A VCG redistribution mechanism is characterized by the function h. Not all h functions are valid, since we need to enforce the weakly budget-balanced constraint.

After simplification, we have that the total payment received by the agents is

$$(n-1)\max\{\sum_i \theta_i, 1\} - \sum_i h(\theta_{-i})$$

We note that $\max\{\sum_i \theta_i, 1\}$ is exactly the first-best total utility. For presentation purposes, we use $S(\theta)$ to denote this expression, where θ is the type profile.

Therefore, a VCG redistribution mechanism characterized by h is weakly budget-balanced if and only if

$$\forall \theta, \quad (n-1)S(\theta) \leq \sum_i h(\theta_{-i})$$

Because all VCG redistribution mechanisms are efficient and strategy-proof, under it, agent i's utility equals $\sum_i \theta_i - h(\theta_{-i})$ if the decision is to build, and her utility equals $1 - h(\theta_{-i})$ if the decision is not to build. That is, agent i's utility equals $S(\theta) - h(\theta_{-i})$. The total utility is then

$$nS(\theta) - \sum_i h(\theta_{-i})$$

The mechanism has a worst-case efficiency ratio α if

$$\forall \theta, \quad nS(\theta) - \sum_i h(\theta_{-i}) \geq \alpha S(\theta)$$

That is,

$$\forall \theta, \quad \sum_i h(\theta_{-i}) \leq (n-\alpha)S(\theta)$$

Combining the above with the weakly budget-balanced constraint, we have

$$\forall \theta, \quad (n-1)S(\theta) \leq \sum_i h(\theta_{-i}) \leq (n-\alpha)S(\theta) \qquad (1)$$

In conclusion, our task is to design h that satisfies Inequality (1) with the aim of maximizing the worst-case efficiency ratio α.

Naroditskiy *et al.* [2012] derived one optimal h function for the case of 3 agents. Guo and Shen [2017] used Automated Mechanism Design technique and identified another optimal solution for 3 agents. We present the latter solution below as one of our results builds on it:

$$h(\theta_{-i}) = T(\theta_{-i}, \frac{2}{3}) + \frac{1}{2}T(\theta_{-i}, 1) - \frac{1}{2}T(\theta_{-i}^1, \frac{2}{3}) - \frac{1}{6} \quad (2)$$

$T(\theta_{-i}, b)$ is defined as the maximum between "the sum of the types from θ_{-i}" and constant b. $T(\theta_{-i}^1, b)$ is defined as the maximum between "the highest type from θ_{-i}" and b.

Unfortunately, neither of the previous results can be generalized to cases with larger numbers of agents. Naroditskiy *et al.* [2012]'s manual technique only works for 3 agents. Guo and Shen [2017]'s Automated Mechanism Design technique can handle more than 3 agents, but like many Automated Mechanism Design approaches, it soon reaches its computational limit around 10 agents. Also, the derived mechanisms are not optimal when the number of agents is above 3. One mechanism we propose in this paper is obtained by extending the mechanism described in Equation (2). Our approach can easily handle thousands of agents (it takes $O(n^3)$ time to calculate the payments) and the achieved worst-case efficiency ratios are also better.

3 Dimension Reduction Technique

The main challenge of designing h is its high dimension when n is large. When $n = 100$, h is a function with 99 variables, which makes it nearly impossible for manual design and computationally too expensive for techniques like Automated Mechanism Design. We have identified an effective technique for reducing the dimension of h, which leads to easy analysis and great worst-case ratios. We can reduce the dimension of h by focusing on the following family of mechanisms:

- We randomly pick 3 distinct agents i, j and k among the n agents.
- If an agent a is not picked, then her additional payment $h(\theta_{-a})$ is set to 0.
- If agent i is picked together with j and k, then her additional payment is set to

  $$h(\theta_{-i}) = g(\theta_j, \theta_k, z_{-ijk})$$

  Here, z_{-ijk} represents the sum of the types from the remaining $n-3$ agents (those that are not i, j, and k). That is, if an agent is picked, then her additional payment depends only on the types of the other two agents who are also picked, and the sum of the types from the remaining $n-3$ agents.

By focusing on the above family of mechanisms, we are essentially dealing with g, which has only 3 variables. By focusing on the above family, we expect to lose on worst-case efficiency ratios (as we are imposing constraints). On the other hand, the design problem is now much easier to handle,

[2] For our objective, it is without loss of generality to focus on anonymous mechanisms.

and our results show that this dimension reduction technique leads to good worst-case efficiency ratios.[3]

We average over all the possible random choices to end up with a mechanism that is deterministic and anonymous. For agent i, there is $3/n$ chance of being picked, we have

$$h(\theta_{-i}) = \frac{3}{n(n-1)(n-2)} \sum_{j,k \neq i} g(\theta_j, \theta_k, z_{-ijk})$$

$$\begin{aligned}\sum_i h(\theta_{-i}) &= \frac{3}{n(n-1)(n-2)} \sum_{i,j,k} g(\theta_j, \theta_k, z_{-ijk}) \\ &= \frac{1}{n(n-1)(n-2)} \sum_{i,j,k} (g(\theta_j, \theta_k, z_{-ijk}) \\ &\quad + g(\theta_i, \theta_k, z_{-ijk}) + g(\theta_i, \theta_j, z_{-ijk}))\end{aligned}$$

For presentation purposes, we introduce function f, which is just g divided by $(n-1)$. We have $\sum_i h(\theta_{-i})$ equals

$$\begin{aligned}&\frac{1}{n(n-2)} \sum_{i,j,k} (f(\theta_j, \theta_k, z_{-ijk}) + f(\theta_i, \theta_k, z_{-ijk}) \\ &\qquad + f(\theta_i, \theta_j, z_{-ijk}))\end{aligned} \tag{3}$$

Recalling Inequality 1, we want the above sum (3) to be at least $(n-1)S(\theta)$ and at most $(n-\alpha)S(\theta)$. Since we want α to be as large as possible, we basically want the above sum to be at least $(n-1)S(\theta)$ and be as close to $(n-1)S(\theta)$ itself as possible. One way to work toward this goal is to design f so that the following is at least $S(\theta)$ and be as close to $S(\theta)$ itself as possible.

$$f(\theta_j, \theta_k, z_{-ijk}) + f(\theta_i, \theta_k, z_{-ijk}) + f(\theta_i, \theta_j, z_{-ijk}) \tag{4}$$

We start with a trivial scenario. If $z_{-ijk} \geq 1$, then obviously the decision is to build (and this signal can be picked up by all of i, j, and k since this signal does not depend on any of their types). In this case, we simply set

$$z \geq 1 \Rightarrow f(a, b, z) = \frac{a+b}{2} + \frac{z}{3} \tag{5}$$

As a result, if $z_{-ijk} \geq 1$, we have

$$\begin{aligned}&f(\theta_j, \theta_k, z_{-ijk}) + f(\theta_i, \theta_k, z_{-ijk}) + f(\theta_i, \theta_j, z_{-ijk}) \\ &= \frac{\theta_j + \theta_k}{2} + \frac{z_{-ijk}}{3} + \frac{\theta_i + \theta_k}{2} + \frac{z_{-ijk}}{3} + \frac{\theta_i + \theta_j}{2} + \frac{z_{-ijk}}{3} \\ &= \theta_i + \theta_j + \theta_k + z_{-ijk} = S(\theta)\end{aligned}$$

Recall that our goal is to make the sum (4) to be above $S(\theta)$ and be as close to $S(\theta)$ as possible. With the setup in (5), we have made the sum exactly $S(\theta)$. So when $z_{-ijk} \geq 1$, the setup in (5) is optimal. We will stick to this setup in our mechanisms.

If for all i, j, k, we have $z_{-ijk} \geq 1$, then by the above setup, we have achieved an efficiency ratio of 1:

$$\sum_i h(\theta_{-i}) = \frac{1}{n(n-2)} n(n-1)(n-2)S(\theta) = (n-1)S(\theta)$$

Lemma 1. *Under the setup specified in* (5)*, the efficiency ratio is* 1 *when the sum of the types is at least* 4*.*

Proof. If $\sum_i \theta_i \geq 4$, then $z_{-ijk} \geq 1$ for all i, j, k, because each type is at most 1. □

4 Asymptotically Optimal Mechanism

Let $I(\text{COND})$ be the characteristic function.

$$I(\text{COND}) = \begin{cases} 1 & \text{COND=TRUE} \\ 0 & \text{COND=FALSE} \end{cases}$$

We first introduce a technical assumption that is needed in our proof.

Definition 2 (Bounded Precision Assumption)**.** *Under the* Bounded Precision Assumption*, every agent's type is a rational number where the denominator is bounded above by constant C.*

As mentioned earlier, the above assumption is not too restrictive. For example, if the agents' valuations are expressed in terms of real-life currency values, then our technical assumption holds. Or if the agents' valuations are expressed using fixed-precision formats (*e.g.*, byte-sized integers[4]), then our assumption holds.

Theorem 1. *Under the* Bounded Precision Assumption*, the VCG redistribution mechanism characterized by the following f is asymptotically optimal. Namely, as n goes to infinity, the mechanism's worst-case efficiency ratio approaches* 1*.*

If $z \geq 1$, then

$$f(a, b, z) = \frac{a+b}{2} + \frac{z}{3}$$

If $z < 1$, then

$$\begin{aligned}f(a, b, z) = &\frac{z}{3} + \frac{\max\{a+b, 1-z\}}{3} \\ &+ \frac{I(a>0)(\max\{a+b, 1-z\} - \max\{b, 1-z\})}{6} \\ &+ \frac{I(b>0)(\max\{a+b, 1-z\} - \max\{a, 1-z\})}{6} \\ &+ \frac{I(a > 0 \text{ AND } b > 0)(1-z)}{3}\end{aligned} \tag{6}$$

The worst-case efficiency ratio equals

$$\alpha = 1 - \frac{4C(4C-1)(3n-2-8C)}{n(n-2)}$$

[3] There are many possible ways to reduce dimensions. For example, we may randomly pick 4 agents and end up dealing with functions with 4 variables. Or we may group the agents in other ways (*i.e.*, three even-sized groups instead of three individuals and a remaining large group). We have tried different dimension reduction methods, some also led to mechanisms with constant worst-case efficiency ratios, but the method we presented here is the one that led to the best results.

[4] Of course, in this case, the project cost should also be expressed as an integer larger than 1.

Before we prove the theorem, we discuss how we derived the proposed mechanism. The high-level process is that we first construct an unanonymous mechanism that has a few properties that we need, and then anonymize it to get the mechanism described in the theorem.

When we randomly pick three agents, we keep track of the pick order. For example, let us say, the pick order is i, then j, then k. We only need to consider cases where $z_{-ijk} < 1$, due to the setup described in (5).

For agent i, we set $f_i(\theta_j, \theta_k, z_{-ijk})$ to be

$$\max\{0 + \theta_j + \theta_k + z_{-ijk}, 1\}$$
$$= z_{-ijk} + \max\{\theta_j + \theta_k, 1 - z_{-ijk}\}$$

This is ensure that if $\theta_i = 0$, then the above amount is exactly $S(\theta)$.

We then move on to agent j, we set $f_j(\theta_i, \theta_k, z_{-ijk})$ to be

$$I(\theta_i > 0)\left(\max\{\theta_i + \theta_k + z_{-ijk}, 1\} - \max\{\theta_k + z_{-ijk}, 1\}\right)$$
$$= I(\theta_i > 0)\left(\max\{\theta_i + \theta_k, 1 - z_{-ijk}\} - \max\{\theta_k, 1 - z_{-ijk}\}\right)$$

The characteristic function ensures that if $\theta_i = 0$, then the above amount is 0 (i has already figured out the exact value of $S(\theta)$, so j should not mess with it). Otherwise, if $\theta_j = 0$, the sum of $f_i(\theta_j, \theta_k, z_{-ijk})$ and $f_j(\theta_i, \theta_k, z_{-ijk})$ is exactly $S(\theta)$.

Finally, for agent k, we set $f_k(\theta_i, \theta_j, z_{-ijk})$ to be

$$I(\theta_i > 0 \text{ AND } \theta_j > 0)(1 - z_{-ijk})$$

The characteristic function ensures that if either θ_i or θ_j equals 0, then the above amount is 0.

If either θ_i or θ_j is 0, then the sum (4) is exactly $S(\theta)$. If neither θ_i nor θ_j is 0, then the sum (4) is between $S(\theta)$ and $2S(\theta)$.[5]

The mechanism proposed in the theorem is obtained by averaging over the $3! = 6$ possible permutations of the three picked agents, in order to make the mechanism anonymous. The anonymized version has two properties that we need: if two or more agents among the three picked agents have type 0, then the sum (4) is exactly $S(\theta)$. Otherwise, the sum (4) is between $S(\theta)$ and $2S(\theta)$.

Now let us prove Theorem 1.

Proof. First of all, we do not need to consider type profiles where the sum of the types is at least 4 due to Lemma 1.

The *Bounded Precision Assumption* assumes that an agent's type has the form p/q where p and q are integers, and $p \le q \le C$. Therefore, the minimal non-zero type is $1/C$. That is, the total number of agents with non-zero types is at most $4C$.

Let us recall Equation 3. $\sum_i h(\theta_{-i})$ equals

$$\frac{\sum_{i,j,k}\left(f(\theta_j, \theta_k, z_{-ijk}) + f(\theta_i, \theta_k, z_{-ijk}) + f(\theta_i, \theta_j, z_{-ijk})\right)}{n(n-2)}$$

If at least two types among θ_i, θ_j, and θ_k are 0s, we have

$$f(\theta_j, \theta_k, z_{-ijk}) + f(\theta_i, \theta_k, z_{-ijk}) + f(\theta_i, \theta_j, z_{-ijk}) = S(\theta)$$

If at most one type is 0, then we have

$$f(\theta_j, \theta_k, z_{-ijk}) + f(\theta_i, \theta_k, z_{-ijk}) + f(\theta_i, \theta_j, z_{-ijk})$$

is between $S(\theta)$ and $2S(\theta)$.

We immediately have that $\sum_i h(\theta_{-i})$ is at least $(n-1)S(\theta)$.

For three random picks, the probability of having at most one 0 is at most

$$p = \frac{4C}{n}\frac{4C-1}{n-1}\frac{4C-2}{n-2} + 3\frac{4C}{n}\frac{4C-1}{n-1}\frac{n-4C}{n-2}$$

Therefore, $\sum_i h(\theta_{-i})$ is at most

$$\frac{n(n-1)(n-2)}{n(n-2)}\left(p2S(\theta) + (1-p)S(\theta)\right)$$
$$= \left((n-1) + \frac{4C(4C-1)(3n-2-8C)}{n(n-2)}\right)S(\theta)$$

Therefore, the worst-case efficiency ratio equals

$$\alpha = 1 - \frac{4C(4C-1)(3n-2-8C)}{n(n-2)}$$

When n goes to infinity, the worst-case efficiency ratio approaches 1. □

Example 1. *Let us consider an example where $C = 10$.*

- *If $C = 10$ and $n = 10000$, then $\alpha = 0.553$.*
- *If $C = 10$ and $n = 100000$, then $\alpha = 0.953$.*
- *If $C = 10$ and $n = 1000000$, then $\alpha = 0.995$.*
- *If $C = 10$ and $n = 10000000$, then $\alpha \approx 1.000$.*

We conclude this section with another asymptotic result.

Theorem 2. *We assume the agents' types are drawn independently and identically from a distribution, with μ and σ being the expectation and the standard deviation, respectively. We assume $\mu > 0$ and $\sigma > 0$. If we use the setup introduced in (5) (true for all mechanisms proposed in this paper), then the efficiency ratio is 1 with probability at least*

$$1 - \frac{n\sigma^2}{(n\mu - 4)^2} \tag{7}$$

When n goes to infinity, the above probability approaches 1.

Proof. Let us consider the sum of the agents' types, denoted by s. The sum's expectation and standard deviation are $\mu' = n\mu$ and $\sigma' = \sqrt{n}\sigma$. When n is large enough, $\mu' = n\mu$ is above 4. The probability that s is less than 4 is at most

$$P(s < 4) \le P(|s - \mu'| \ge (\mu' - 4)) = P(|s - \mu'| \ge \sigma'\frac{\mu' - 4}{\sigma'})$$

This is at most (due to Chebyshev's inequality)

$$\frac{\sigma'^2}{(\mu' - 4)^2} = \frac{n\sigma^2}{(n\mu - 4)^2}$$

Since μ and σ are constants, when n goes to infinity, the probability of $s < 4$ approaches 0. When the sum of the types s is at least 4, our mechanism's efficiency ratio is 1 according to Lemma 1. Hence, asymptotically, our mechanism's efficiency ratio is 1 with probability 1. □

[5] This can be established via algebraic simplification, so we omit the details due to space constraint.

Example 2. *Let us consider an example where the agents' types are drawn independently and identically from a uniform distribution $U(0,1)$. Here, $\mu = 1/2$ and $\sigma^2 = 1/12$. According to Equation 7, under our mechanism, the efficiency ratio is 1 with probability at least $1 - \frac{n}{3(n-8)^2}$.*

- *If $n = 20$, the probability is at least 0.954.*
- *If $n = 50$, the probability is at least 0.991.*
- *If $n = 100$, the probability is at least 0.996.*

5 Extending the Optimal Mechanism for Three Agents

In this section, we focus on smaller numbers of agents and derive a mechanism that achieves better worst-case efficiency ratios than previous results.

We extend the optimal mechanism from [Guo and Shen, 2017]. We use h^{3*} to denote the optimal h function when $n = 3$. h^{3*} from Equation (2) can be rewritten as $h^{3*}(a,b) =$

$$\max\{a+b, \frac{2}{3}\} + \frac{1}{2}\max\{a+b,1\} - \frac{1}{2}\max\{a,b,\frac{2}{3}\} - \frac{1}{6}$$

Lemma 2. *(Due to [Guo and Shen, 2017]): For any $0 \le a, b, c \le 1$, we have*

$$2\max\{a+b+c,1\} \le h^{3*}(a,b) + h^{3*}(b,c) + h^{3*}(a,c)$$
$$\le 7/3\max\{a+b+c,1\}$$

Next, we modify h^{3*} slightly into $h^{3*'}$ (replacing a, b by $\min\{a,1\}$ and $\min\{b,1\}$, respectively, and then added $a - \min\{a,1\} + b - \min\{b,1\}$):

$$\begin{aligned} h^{3*'}(a,b) = {} & a - \min\{a,1\} + b - \min\{b,1\} \\ & + \max\{\min\{a,1\} + \min\{b,1\}, \frac{2}{3}\} \\ & + \frac{1}{2}\max\{\min\{a,1\} + \min\{b,1\}, 1\} \\ & - \frac{1}{2}\max\{\min\{a,1\}, \min\{b,1\}, \frac{2}{3}\} - \frac{1}{6} \end{aligned}$$

Lemma 3. *For any $0 \le a, b, c$, we have*

$$2\max\{a+b+c,1\} \le h^{3*'}(a,b) + h^{3*'}(b,c) + h^{3*'}(a,c)$$
$$\le 7/3\max\{a+b+c,1\}$$

The main difference between Lemma 2 and Lemma 3 is that we no longer require that $a, b, c \le 1$.

Finally, for all constant terms mentioned in the description of $h^{3*'}$, we can scale it by a factor of t and maintain a similar inequality to that in Lemma 3. We modify $h^{3*'}$ into $h^{3*''}$ by scaling the constants by a factor of t:

$$\begin{aligned} h^{3*''}(a,b,t) = {} & a - \min\{a,t\} + b - \min\{b,t\} \\ & + \max\{\min\{a,t\} + \min\{b,t\}, \frac{2t}{3}\} \\ & + \frac{1}{2}\max\{\min\{a,t\} + \min\{b,t\}, t\} \\ & - \frac{1}{2}\max\{\min\{a,t\}, \min\{b,t\}, \frac{2t}{3}\} - \frac{t}{6} \end{aligned}$$

Lemma 4. *For all $0 \le a, b, c$ and $0 < t$, we have*

$$2\max\{a+b+c,t\} \le h^{3*''}(a,b,t) + h^{3*''}(b,c,t) + h^{3*''}(a,c,t)$$
$$\le 7/3\max\{a+b+c,t\}$$

Theorem 3. *The VCG redistribution mechanism characterized by the following f has a worst-case efficiency ratio of $\frac{n+1}{2n}$.*

If $z \ge 1$, then

$$f(a,b,z) = \frac{a+b}{2} + \frac{z}{3}$$

If $z < 1$, then

$$f(a,b,z) = \frac{z}{3} + \frac{h^{3*''}(a,b,1-z)}{2}$$

Proof. We have that for all $0 \le \theta_i, \theta_j, \theta_k$ and $0 \le z_{-ijk} < 1$,

$$f(\theta_j,\theta_k,z_{-ijk}) + f(\theta_i,\theta_k,z_{-ijk}) + f(\theta_i,\theta_j,z_{-ijk})$$
$$= z_{-ijk} + \frac{h^{3*''}(\theta_j,\theta_k,1-z_{-ijk})}{2}$$
$$+ \frac{h^{3*''}(\theta_i,\theta_k,1-z_{-ijk})}{2} + \frac{h^{3*''}(\theta_i,\theta_j,1-z_{-ijk})}{2}$$

Therefore, we have

$$S(\theta) = \max\{\theta_i + \theta_j + \theta_k, 1 - z_{-ijk}) + z_{-ijk} \le$$
$$f(\theta_j,\theta_k,z_{-ijk}) + f(\theta_i,\theta_k,z_{-ijk}) + f(\theta_i,\theta_j,z_{-ijk})$$
$$\le 7/6\max\{\theta_i + \theta_j + \theta_k, 1 - z_{-ijk}) + z_{-ijk}$$

For the mechanism characterized by the above f function, $\sum_i h(\theta_{-i})$ is at least $(n-1)S(\theta)$, as required by Inequality 1. Next we analyze how large $\sum_i h(\theta_{-i})$ can be.

We first consider type profiles where the decision is to build. $\sum_i h(\theta_{-i})$ is at most

$$\frac{1}{n(n-2)}\sum_{i,j,k}(7/6\max\{\theta_i + \theta_j + \theta_k, 1 - z_{-ijk}) + z_{-ijk})$$
$$= (n-1)S(\theta) + \frac{n-1}{2n}S(\theta) = (n - \frac{n+1}{2n})S(\theta)$$

So when the decision is to build, the corresponding worst-case efficiency ratio is $\frac{n+1}{2n}$. We omit the case where the decision is not to build due to space constraint.

□

In summary, for the mechanism we introduced in this section, which does not require the *Bounded Precision Assumption*, the worst-case efficiency ratio is $\frac{n+1}{2n}$. Asymptotically, this is 0.5. We have included this mechanism in the table in Subsection 1.2. This mechanism outperforms all previously proposed mechanisms in terms of worst-case efficiency ratios (both asymptotically and for smaller number of agents). The mechanism's implementation takes $O(n^3)$ complexity, as we need to average over all possible draws of three agents.

References

[Cavallo, 2006] Ruggiero Cavallo. Optimal decision-making with minimal waste: Strategyproof redistribution of vcg payments. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 882–889, New York, NY, USA, 2006. ACM.

[Gujar and Narahari, 2011] Sujit Gujar and Y. Narahari. Redistribution mechanisms for assignment of heterogeneous objects. *J. Artif. Intell. Res.*, 41:131–154, 2011.

[Guo and Conitzer, 2009] Mingyu Guo and Vincent Conitzer. Worst-case optimal redistribution of VCG payments in multi-unit auctions. *Games and Economic Behavior*, 67(1):69–98, 2009.

[Guo and Shen, 2017] Mingyu Guo and Hong Shen. Speed up automated mechanism design by sampling worst-case profiles: An application to competitive VCG redistribution mechanism for public project problem. In Bo An, Ana L. C. Bazzan, João Leite, Serena Villata, and Leendert W. N. van der Torre, editors, *PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference, Nice, France, October 30 - November 3, 2017, Proceedings*, volume 10621 of *Lecture Notes in Computer Science*, pages 127–142. Springer, 2017.

[Guo *et al.*, 2011] Mingyu Guo, Victor Naroditskiy, Vincent Conitzer, Amy Greenwald, and Nicholas R. Jennings. Budget-balanced and nearly efficient randomized mechanisms: Public goods and beyond. In Ning Chen, Edith Elkind, and Elias Koutsoupias, editors, *Internet and Network Economics - 7th International Workshop, WINE 2011, Singapore, December 11-14, 2011. Proceedings*, volume 7090 of *Lecture Notes in Computer Science*, pages 158–169. Springer, 2011.

[Guo *et al.*, 2013] Mingyu Guo, Evangelos Markakis, Krzysztof R. Apt, and Vincent Conitzer. Undominated groves mechanisms. *J. Artif. Intell. Res.*, 46:129–163, 2013.

[Guo, 2011] Mingyu Guo. VCG redistribution with gross substitutes. In Wolfram Burgard and Dan Roth, editors, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. AAAI Press, 2011.

[Guo, 2012] Mingyu Guo. Worst-case optimal redistribution of VCG payments in heterogeneous-item auctions with unit demand. In Wiebe van der Hoek, Lin Padgham, Vincent Conitzer, and Michael Winikoff, editors, *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*, pages 745–752. IFAAMAS, 2012.

[Guo, 2016] Mingyu Guo. Competitive VCG redistribution mechanism for public project problem. In Matteo Baldoni, Amit K. Chopra, Tran Cao Son, Katsutoshi Hirayama, and Paolo Torroni, editors, *PRIMA 2016: Princiles and Practice of Multi-Agent Systems - 19th International Conference, Phuket, Thailand, August 22-26, 2016, Proceedings*, volume 9862 of *Lecture Notes in Computer Science*, pages 279–294. Springer, 2016.

[Holmström, 1979] Bengt Holmström. Groves' scheme on restricted domains. *Econometrica: Journal of the Econometric Society*, pages 1137–1144, 1979.

[Manisha *et al.*, 2018] Padala Manisha, C. V. Jawahar, and Sujit Gujar. Learning optimal redistribution mechanisms through neural networks. In Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar, editors, *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 345–353. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.

[Mas-Colell *et al.*, 1995] Andreu Mas-Colell, Michael Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[Moore, 2006] J. Moore. *General Equilibrium and Welfare Economics: An Introduction*. Springer, 2006.

[Moulin, 1988] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

[Moulin, 2009] Hervé Moulin. Almost budget-balanced VCG mechanisms to assign multiple objects. *JET*, 144(1):96–119, 2009.

[Naroditskiy *et al.*, 2012] Victor Naroditskiy, Mingyu Guo, Lachlan Dufton, Maria Polukarov, and Nicholas R. Jennings. Redistribution of VCG payments in public project problems. In Paul W. Goldberg, editor, *Internet and Network Economics - 8th International Workshop, WINE 2012, Liverpool, UK, December 10-12, 2012. Proceedings*, volume 7695 of *Lecture Notes in Computer Science*, pages 323–336. Springer, 2012.