Les Cahiers du GERAD ISSN: 0711–2440

A heuristic optimization of Bayesian incentive-compatible cake-cutting

L.N. Hoang, F. Soumis,
G. Zaccour

G–2014–80

November 2014

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A heuristic optimization of Bayesian incentive-compatible cake-cutting

Lê Nguyên Hoang [a]
François Soumis [a]
Georges Zaccour [b]

[a] *GERAD & Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7*

[b] *GERAD & HEC Montréal, Montréal (Québec) Canada, H3T 2A7*

len.hoang.lnh@gmail.com
francois.soumis@gerad.ca
georges.zaccour@gerad.ca

November 2014

Les Cahiers du GERAD
G–2014–80

Abstract: Cake-cutting is a metaphor for problems where a principal agent has to fairly allocate resources. Such problems cover various areas of operations research and management science, like, for instance, shift scheduling with employees' preferences. Recent work focuses on optimizing social efficiency while guaranteeing fairness, but ignore incentive-compatibility constraints, or vice versa. In this paper, we present a new approach to heuristic mechanism design with Bayesian incentive-compatibility. As opposed to other papers, we do not allow monetary transfer. Our approach relies on the revelation principle and the computation of Bayesian-Nash equilibria using the so-called return function. This computation consists in tracking a best-reply dynamics of return function, which are mappings of action to probability distribution on outcomes, instead of the more classical but harder-to-compute best-reply dynamics of strategies. In essence, our mechanism-design approach explores a parameterized class of revelation mechanisms, which we know by construction to be Bayesian incentive-compatible. We highlight the efficiency of this approach through numerical results on instances of respectively 2, 5 and 20 agents.

Key Words: Cake-cutting problem; fairness; incentive compatibility; return function.

Résumé : Le partage de gâteau est une métaphore très utilisée pour décrire les problèmes où un agent principal doit allouer des ressources de manière juste. De tels problèmes couvrent une grande variété de domaines de recherche opérationnelle, comme, par exemple, la construction d'horaires personnalisés avec les préférences des employés. Plusieurs travaux récents cherchent à maximiser le bien-être social tout en garantissant l'équité (Cohler et al. (2001), Caragiannis et al. (2011), Bei et al. (2012)), mais ignorent les contraintes de compatibilité avec les incitatifs. À l'inverse, Mossel et Tamuz (2010) et Chen et al. (2013) s'intéressent à l'équité et les incitatifs sans considérer le bien-être social. Dans ce papier, nous présentons une approche pour optimiser un compromis d'équité et de bien-être social, tout en satisfaisant les contraintes d'incitatifs bayésiens. Cette approche repose sur le principe de révélation de Gibbard (1973) et Myerson (1979), et sur le calcul des équilibres de Bayes-Nash proposé par Hoang et al. (2014). Ce calcul consiste à simuler une dynamique de meilleures réponses de fonctions de retour, qui associent à toute action une distribution probabiliste de résultats, plutôt que de simuler une dynamique de meilleures réponses des stratégies, comme cela est fait de façon plus classique. On peut alors explorer une classe de mécanismes révélés paramétrés, qui, par construction, sont compatibles avec les incitatifs bayésiens. Nous montrons l'efficacité de l'approche avec des instances de 2, 5 et 20 joueurs.

1 Introduction

In this paper, we propose a mechanism to divide a resource among a set of agents who have heterogenous preferences and may not want to reveal their preferences truthfully. We suppose that the mechanism is designed by a principal agent (or mechanism designer), whose aim is to achieve a specified objective, e.g., maximizing a weighted sum of collective efficiency and fairness indicators. Our contribution belongs to the vast literature dealing with the cake-cutting problem, which is considered a dynamic and challenging field of investigation in operations research and management science (see, e.g., Brams and Taylor (1996); Robertson and Webb (1998); Mossel and Tamuz (2010); Chen et al. (2013)).

The cake-cutting problem, which was first introduced in Steinhaus (1948), consists in devising a method to fairly allocate a cake to a set of agents. Many procedures have been proposed over time to do so, including the well known *last diminisher method* Steinhaus (1948), the *divide-and-choose* approach, the *moving knife procedure* Brams and Taylor (1996), and *the successive pairs algorithm* Robertson and Webb (1998). These mechanisms share the property of being weakly incentive-compatible, that is, an untruthful claimer may regret her untruthfulness at some point. However, if one requires a stronger concept of incentive-compatibility, e.g., dominant strategy incentive-compatibility (DSIC), meaning that agents always have incentives to be truthful, then the problem may end up having no conceptually satisfying solution. To illustrate, Mossel and Tamuz (2010) proved that there exists no deterministic DSIC super-fair division to the cake-cutting problem. We recall that a super-fair division yields the exact (or proportional) division solution when all players have the same preferences, and does strictly better than exact division otherwise. Recently, Chen et al. (2013) added the assumption that players have piecewise constant valuation functions, and provided a proportionally fair and envy-free deterministic DSIC mechanism. A randomized DSIC super-fair division is discussed in Mossel and Tamuz (2010) and Chen et al. (2013).

In the above-cited studies, little attention has been given to social efficiency, which could be a legitimate objective. Some recent work focuses on optimizing social efficiency while guaranteeing fairness, but ignore incentive-compatibility constraints (see, e.g., Caragiannis et al. (2011); Bei et al. (2012)). In Cohler et al. (2011), the authors provided a tractable, nearly optimal envy-free mechanism when the agents truthfully report their valuations of the cake. They also gave optimal envy-free mechanisms for certain specific structures of the agents' preferences. Note that the difficulty in determining a socially efficient solution is due to the assumptions that (i) the cake is divisible into an infinite number of portions, and (ii) the agents' utility functions are complex mathematical objects involving probability measures. Here, we assume that the cake is made of homogeneous portions, with the agents having constant valuations over each of these portions. In particular, this will enable us to write the set of admissible allocations as a polytope.

Our approach has its roots in mechanism-design theory, where one retains Bayesian incentive-compatibility (BIC) instead of DSIC. This means that we require truthfulness to be a Bayesian-Nash equilibrium. In other words, assuming others truthful, it is in every agent's best interest to reveal her preferences truthfully. One popular approach for BIC design introduced be Myerson (1981) for the context of auctions relies on so-called agents' virtual valuations. This approach has gained recent interests through developments of *ironing* methods in multi-dimensional auctions (see, e.g., Parkes (2009); Bei and Huang (2011); Hartline and Lucier (2010); Hartline et al. (2011); Hartline (2013)). However, such approaches strongly rely on monetary transfers and agents' risk neutrality.

The main contribution of this paper is a general approach to BIC mechanism design, which requires none of these assumptions: We do not allow monetary transfers nor assume risk neutrality. Instead, we will rely on the revelation principle introduced in Gibbard (1973); Myerson (1979, 1981). Applying this revelation principle requires the computation of a Bayesian-Nash equilibrium. To do so, we implement the algorithm proposed in Hoang et al. (submitted), which uses the concept of return functions. (We wish to mention from the outset that there is a huge difference between this paper and Hoang et al. (submitted), namely, here the focus is on selecting or designing a mechanism, whereas in Hoang et al. (submitted) no such issue is involved as the mechanism is given.) A return function is a mapping of an agent's action to the induced probability distribution on her outcomes. The authors showed that any strategy profile in a Bayesian game generates a return-function profile that captures all the information required to describe the best-reply dynamics. Consequently, any best-reply dynamics of strategies is naturally mapped to a best-reply dynamics of return functions. It is significant that return functions and best-replies to return functions are much easier to compute than strategies and best-replies to strategies. This advantage is particularly valuable when the beliefs or the Bayesian game cannot be described analytically, as will be the case in our cake-cutting problem. In a second step, we apply the revelation principle to obtain a BIC mechanism. Finally, we compute the value of the mechanism designer's objective function. This is done iteratively, where at each iteration, the principal draws types of players according to beliefs.

We shall apply this general approach to a cake-cutting problem that aims at a balance of social efficiency and fairness. For this problem, we present the principal's optimal mechanism in the case where she has knowledge of players' preferences, which we call the ideal mechanism. Then, we show that this ideal mechanism performs poorly when players play a Bayesian-Nash equilibrium through numerical simulations. Third, we will provide a family of quality parameterized cake-cutting mechanisms to which we shall apply our general approach to BIC mechanism design. The computed results quantify the principal's objective values for the revelations of our family of parameterized mechanisms, which turn out to be significantly improving on the revelation of the ideal mechanism.

The rest of the paper is organized as follows: In Section 2, we provide the theoretical foundations to our approach by describing a general setting of mechanism design and by introducing the return function. In Section 3, we present algorithms for computing Bayesian-Nash equilibria, implementing the revelation principle and optimizing the mechanism designer's objective. In Section 4, we provide an illustrative example of a cake-cutting problem and report the computational results. Finally, we conclude in Section 5.

2 General model

Denote by $N = \{1, \dots, n\}$ the set of players participating in the cake-sharing problem. The mechanism designer (or principal) in charge of dividing the cake seeks an allocation that optimizes a given criterion specified below. Denote by x_i the share of player i, with $x_i \in X_i$, and by θ_i the type of player i, with $\theta_i \in \Theta_i$. Player i's preferences are described by a utility function that depends only on these data, namely, $u_i(\theta_i, x_i) \in \mathbb{R}$. We assume that the principal and all players but i have the same incomplete information about player i's type θ_i. This incomplete information is described by a probability distribution, called belief, $\tilde{\theta}_i \in \Delta(\Theta_i)$ on player i's type. We suppose that the beliefs about the players' types are independent.

2.1 Direct mechanisms

The principal would ideally choose (possibly stochastically) an allocation $x = (x_1, \dots, x_n) \in X$ with the knowledge of the type profile $\theta = (\theta_1, \dots, \theta_n) \in \Theta$. Such a choice is known as a direct mechanism.

Definition 1 *A direct mechanism $\mathcal{D}$ is a mapping of type profiles $\theta \in \Theta$ into probability distributions on outcomes $\mathcal{D}(\theta) \in \Delta(X)$, i.e., $\mathcal{D} : \Theta \to \Delta(X)$. We denote by $\mathbb{D}$ the set of direct mechanisms.*

The principal's payoff is defined by the function $\mathcal{P} : \mathbb{D} \to \mathbb{R}$, and her optimization (or mechanism design) problem consists in maximizing $\mathcal{P}$ over a set of incentive-compatible direct mechanisms. The objective function may depend on the belief $\tilde{\theta} = (\tilde{\theta}_1, \dots, \tilde{\theta}_n)$. For instance, the objective function could combine social efficiency and fairness as follows:

$$\mathcal{P}(\mathcal{D}) = \mathbb{E}_{\theta \sim \tilde{\theta}} \Big[\mathbb{E}_{x \sim \mathcal{D}(\theta)} \big[\sum_{i \in N} u_i(\theta_i, x_i) - \lambda Var\{u_i(\theta_i, x_i)\}_{i \in N} \big] \Big], \tag{1}$$

where Var is some measure of the variations among players' utilities, and λ is a nonnegative scaling parameter. The specific structure of the principal's objective function enables us to compute it through a basic Monte-Carlo method, and we will use classical statistical tools to compute a confidence interval.

The concept of incentive compatibility means, among other things, that a mechanism-design problem is a Bayesian game. Indeed, when players learn about the direct mechanism that the principal wants to implement, they may choose to reveal their preferences untruthfully. We distinguish revealed preferences from intrinsic preferences $\theta \in \Theta$ by calling them actions. An action profile is denoted by $a = (a_1, \ldots, a_n) \in \Theta$.

A (mixed) strategy s_i for player i is a mapping from the set Θ_i into $\Delta(\Theta_i)$, i.e., $s_i : \Theta_i \to \Delta(\Theta_i)$. Denoting by $s = (s_1, \ldots, s_n) \in S$ a strategy profile where $S = \Pi_{i \in N} S_i$ and S_i is the set of mixed strategies of player i, we can define the utility of that player in its classical form, that is, in terms of strategies as follows:

$$u_i(s_i, s_{-i}) = \mathbb{E}_{\tilde{\theta}}\Big[\mathbb{E}_{x_i \sim \mathcal{D}_i(s(\theta))}\big[u_i(\theta_i, x_i)\big]\Big], \tag{2}$$

where s_{-i} is the strategy profile of all players but i. The truthful strategy s_i^{truth} of player i is then an identity $s_i(\theta_i) = \theta_i$ for all $\theta_i \in \Theta_i$, where we have identified θ_i with the Dirac distribution $\delta_{\theta_i} \in \Delta(\Theta_i)$.

Equation 2 underlines how the direct mechanism $\mathcal{D}$ defines a Bayesian game. A Bayesian-Nash equilibrium of that game is then a strategy profile s^{BN} such that every s_i^{BN} is a best reply to s_{-i}^{BN}. This leads us to the definition of Bayesian incentive-compatibility.

Definition 2 *A direct mechanism $\mathcal{D} \in \mathbb{D}$ is Bayesian incentive-compatible (BIC) if truthfulness is a Bayesian-Nash equilibrium. We denote by $\mathbb{D}_{BIC}$ the set of BIC mechanisms.*

The Bayesian-mechanism-design problem can then be stated as follows:

$$\mathcal{P}^*_{BIC} = \sup_{\mathcal{D}_{BIC} \in \mathbb{D}_{BIC}} \mathcal{P}(\mathcal{D}_{BIC}). \tag{3}$$

We will use the term BIC-optimal value for the optimal value $\mathcal{P}^*_{BIC}$ of the Bayesian-mechanism-design problem. This value could be compared to the ideal value that the mechanism designer could achieve if she had complete information about players' types. This ideal value is given by

$$\mathcal{P}^*_{ideal} = \sup_{\mathcal{D} \in \mathbb{D}} \mathcal{P}(\mathcal{D}). \tag{4}$$

Note that the BIC requirement can be regarded as adding an infinite number of incentive-compatibility constraints of the form

$$\forall s_i \in S_i, \quad u_i(s_i^{truth}, s_{-i}^{truth}) \geq u_i(s_i, s_{-i}^{truth}). \tag{5}$$

The ideal value is then obtained by relaxing the Bayesian-mechanism-design problem, and therefore $\mathcal{P}^*_{ideal}$ yields an upper bound to the BIC-optimal value $\mathcal{P}^*_{BIC}$. Clearly, $\mathcal{P}^*_{ideal}$ is in general easier to compute than $\mathcal{P}^*_{BIC}$.

2.2 Revelation principle

The focus on direct mechanisms instead of say, more sophisticated ones in terms of complex sets of actions, is reminiscent of the revelation principle. Given a mechanism $\mathcal{D}$ and a Bayesian-Nash equilibrium strategy profile s^{BN}, we can construct the revelation mechanism $\mathcal{D}_{Rev} : \Theta \to \Delta(X)$ defined by

$$\mathcal{D}_{Rev}(\theta) = \mathcal{D}(s^{BN}(\theta)). \tag{6}$$

In other words, $\mathcal{D}_{Rev}$ is merely the function composition $\mathcal{D} \circ s^{BN}$. Note that the revelation mechanism does not require the initial mechanism to be direct, but we do not make the general case explicit, for the sake of expository clarity. What is important is that we have the following theorem.

Theorem 3 *The set of revelation mechanisms coincides with $\mathbb{D}_{BIC}$.*

Sketch of proof. Consider a BIC mechanism $\mathcal{D}_{BIC}$. Since truthfulness is a Bayesian-Nash equilibrium of $\mathcal{D}_{BIC}$, the mechanism $\mathcal{D}_{Rev} = \mathcal{D}_{BIC} \circ s^{truth}$ is a revelation mechanism. Yet, it equals $\mathcal{D}_{BIC}$. Thus, any BIC mechanism is a revelation mechanism.

Reciprocally, assume $\mathcal{D}_{Rev}$ is the revelation mechanism of some direct mechanism $\mathcal{D}$ with some Bayesian-Nash equilibrium s^{BN}. Then, for any player i, playing s_i against s_{-i}^{truth} in $\mathcal{D}_{Rev}$ is equivalent to playing $s_i^{BN} \circ s_i$ against s_{-i}^{BN} in $\mathcal{D}$. As by definition s_i^{BN} is a best reply to s_{-i}^{BN} in $\mathcal{D}$, then $s_i = s_i^{truth}$ is a best reply to s_{-i}^{truth} in $\mathcal{D}_{Rev}$. This shows that $\mathcal{D}_{Rev}$ is BIC and completes the proof. □

The most important message of this theorem is that any mechanism boils down to a (BIC) direct mechanism. This is why the mechanism-design literature has by large focused on direct mechanisms. However, as discussed further, our new approach to Bayesian mechanism design makes greater use of the revelation principle. Indeed, it consists in exploring a parameterized set of direct mechanisms $\mathcal{D}^t$ for some parameter t. For each value of t, we compute Bayesian-Nash equilibria s^t of the game induced by $\mathcal{D}^t$, and we apply the revelation principle to obtain $\mathcal{D}^t_{Rev} = \mathcal{D}^t \circ s^t$. We then search for an optimum among revelation mechanisms $\mathcal{D}^t_{Rev}$ for $t \in T$, which we know by construction to be BIC.

2.3 Return function

A major step in applying the approach outlined above is the computation of a Bayesian-Nash equilibrium s^{BN} of a direct mechanism $\mathcal{D}$. Clearly, there may be no analytical way to do so for complex mechanisms. We propose to use the method introduced by Hoang et al. (submitted), where the key feature is the introduction of the return function.

Definition 4 *A return function $\varphi_i : \Theta_i \to \Delta(X_i)$ for a player i is a mapping of her action a_i with the induced probability distribution $\varphi_i(a_i) \in \Delta(X_i)$ on herallocation. We denote by Φ_i the set of return functions of player i and Φ the set of return-function profiles.*

Crucially, a strategy profile s determines a return-function profile, which is defined by

$$\varphi_i(a_i) := \mathbb{E}_{\theta_{-i} \sim \tilde{\theta}_{-i}} \Big[\mathcal{D}(a_i, s_{-i}(\theta_{-i})) \Big]. \tag{7}$$

Intuitively, the return function φ_i for player i maps player i's action to the probability distribution on the outcome when other players use strategy s_{-i} and when the outcome is determined by the mechanism $\mathcal{D}$, by averaging out all scenarios $\theta_{-i} \sim \tilde{\theta}_{-i}$.

In Hoang et al. (submitted), the authors proved that strategy profiles could be substituted by return-function profiles in order to study best-reply dynamics and Bayesian-Nash equilibria. This means that we have a best-reply correspondence of return-function profiles, which is a simple translation of the best-reply correspondence of strategy profiles. Similarly, Bayesian-Nash strategy profiles are naturally mapped into Bayesian-Nash return-function profiles. Interestingly, the computations of (approximated) return-function profiles are more natural and relevant than the computations of (approximated) strategy profiles for the following reasons. First, unlike strategy profiles, return function profiles are observable objects because one can infer approximated return functions by looking at how actions have affected outcomes. Second, it is more relevant to a player to interpolate return functions to estimate outcomes for unobserved actions than to interpolate others' strategies. Third, computationally, it is easier for a player to determine her best action if given her return function than if given knowledge of the other players' strategies. Indeed, even if player i knows s_{-i}, she still needs to know the beliefs $\tilde{\theta}_{-i}$ and the mechanism $\mathcal{D}$ in order to compute her best reply.

Based on these remarks, Hoang et al. (submitted) designed an algorithm for computing Bayesian-Nash equilibria, where the core idea is to track a sort of fictitious play dynamics in the space of return-function profiles to search for a fixed point. This fixed point corresponds to a Bayesian-Nash return-function profile, which in turn, correspond to a Bayesian-Nash strategy profile.

2.4 Fictitious play

The algorithm in Hoang et al. (submitted) for computing Bayesian-Nash equilibria follows a fictitious-play-dynamics learning approach for the return function. As iterations pile up, players are best-replying to the *estimated* return function, and accumulating knowledge about the best-reply return function.

Algorithm 1: Computation of Bayesian-Nash equilibria by fictitious play

```
Data: belief θ̃, mechanism 𝒟.
Result: Bayesian-Nash return-function profile φ.
iter ← 0;
initialize the return function φ;
while a convergence criterion is not met do
    (1) increment iter;
    for all players i ∈ N do
        (2) draw a type θ_i ~ θ̃_i;
        (3) compute action a_i which maximizes u_i(θ_i, φ_i(a_i));
    end
    (4) draw an outcome x ~ 𝒟(a);
    (5) update the return-function profile φ with iter, a and x;
end
return φ;
```

Note that points (2) and (4) are straightforward uses of the inputs of the algorithm. This means that they are as complicated as the belief and mechanism are. In some applications, e.g., shift scheduling with employees' preferences, these points, especially point (4), may already be quite hard on their own.

However, for our purpose here, the difficult points are rather points (3) and (5), as they require an implementation of the return function. Let us start with point (5), which corresponds to an updating procedure. Given that each return function is built by experience, we propose a simple update, which consists in adding the 3-tuples $(iter, a_i, x_i)$ to the list of observations made so far by player i. Mathematically, this list can be represented by a finite set $List(\varphi_i) \subset \mathbb{N} \times \Theta_i \times X_i$.

In our example of Section 3, because both the beliefs and the mechanism are symmetric (which means that the players' labels are irrelevant), we have $\Theta_i = \Theta_j = \Theta_0$ and $X_i = X_j = X_0$ for all $i, j \in N$. Plus, since we will search for a symmetric Bayesian-Nash equilibrium, we consider a single return function for all players, whose list of observations is then a finite set $List(\varphi) \subset \mathbb{N} \times \Theta_0 \times X_0$. Now, given a list $List(\varphi_i)$ of observations, we still need to compute a value for $u_i(\theta_i, \varphi_i(a_i))$ for any $\theta_i \in \Theta_i$ and $a_i \in \Theta_i$. To do so, we propose the following simple radius-based function interpolation:[1]

$$u_i(\theta_i, \varphi_i(a_i)) = \frac{\sum_{(iter, \hat{a}_i, \hat{x}_i) \in List(\varphi_i)} w_i(a_i, \hat{a}_i, iter)\, u_i(\theta_i, \hat{x}_i)}{\sum_{(iter, \hat{a}_i, \hat{x}_i) \in List(\varphi_i)} w_i(a_i, \hat{a}_i, iter)}, \tag{8}$$

where $w : \Theta_i \times \Theta_i \times \mathbb{N} \to \mathbb{R}$ is some weighting of observations. Typically, the function $w(a_i, \hat{a}_i, iter)$ should be decreasing in the distance between a_i and $\hat{a}_i$, and it could be slowly increasing in $iter$.[2] We refer to Hoang et al. (submitted) for further discussions on this weighting.

Now, even with a good interpolation, point (3) of Algorithm 1 remains difficult, as it involves an optimization problem. In our example, we will solve it using a basic TABU search.

[1] Arguably, there may be better ways to estimate $u_i(\theta_i, \varphi_i(a_i))$, especially if we take into account the structure of the Bayesian game. However, in this paper, the focus is not on the performance of our computation of Bayesian-Nash equilibria.

[2] To illustrate, we will use $w(a_i, \hat{a}_i, iter) = iter/(1 + d_{\Theta_i}(a_i, \hat{a}_i))^2$ in our illustrative example, where d_{Θ_i} is some metric on Θ_i.

3 Application to cake-cutting problem

We illustrate the theory and the algorithm we have developed with a cake-cutting problem. Essentially, one is looking for an allocation of some (possibly heterogenous) cake to different agents having heterogenous and secret preferences.

3.1 Model

We consider a simplified setting where the cake is made of K homogeneous portions, which we call attributes.[3] An allocation $x_i \in X_0$ to player i is then a vector $(x_{i1}, \dots, x_{iK}) \in [0,1]^K$, where each entry x_{ik} stands for the ratio of attribute k that is given to player i. Naturally, the set of outcomes is then given by

$$X = \Big\{ x \in [0,1]^{n \times K} \mid \forall k, \sum_{i \in N} x_{ik} \leq 1 \Big\}. \tag{9}$$

As in most cake-cutting settings, we assume that the utility function of player $i, i \in N$, is additive, that is,

$$u_i(\theta_i, x_i) = \sum_{k=1}^{K} \theta_{ik} x_{ik}, \tag{10}$$

where θ_{ik} is the utility of that player for the whole portion k. Also, as it is classically done, we normalize types θ_i so that the utility of the whole cake equals 1. Consequently, we have $\Theta_0 = \{\theta_0 \in [0,1]^K \mid \sum_{k=1}^{K} \theta_{0k} = 1\}$.

The belief $\tilde{\theta}_i$ of any player i we consider is the uniform distribution over the simplex of $\mathbb{R}^K$. It is well-known that this Dirichlet distribution can be obtained by drawing K i.i.d. exponential variables and normalizing them so that they belong to the simplex. Finally, for a type profile θ associated to an outcome x, we define the principal's objective function as follows:

$$p(\theta, x) = n\bar{u}(\theta, x) - \lambda \sum_{i \in N} |u_i(\theta_i, x_i) - \bar{u}(\theta, x)|, \tag{11}$$

where: $p(\theta, x)$ is the expectation for $\theta \sim \tilde{\theta}$ and $x \sim \mathcal{D}(\theta)$; $\bar{u}(\theta, x) = \frac{1}{n} \sum_{i \in N} u_i(\theta_i, x_i)$ is the average utility, and hence, $n\bar{u}(\theta, x)$ is the collective total utility; and parameter $\lambda \geq 0$ is the weight of fairness in the objective function.

Interestingly, a mechanism allocating $x_{ik} = 1/n$ to each player, $k = 1, \dots, K$, has a principal's objective value of 1, which can be shown to be the optimal mechanism, when the players are not involved (i.e., do not express their preferences).

3.2 Ideal mechanism

Let us consider the direct mechanism $\mathcal{D}^{ideal}$, which inputs a type profile $\theta \in \Theta$ and outputs a solution to the following linear program:

$$\begin{aligned} \underset{x,u,\bar{u},\delta,\gamma}{\text{Maximize}} \quad & n\bar{u} - \lambda \sum_{i \in N} |u_i - \bar{u}|, \\ \text{subject to :} \quad & u_i = \sum_{k=1}^{K} \theta_{ik} x_{ik}, \qquad \forall i \in N, \\ & n\bar{u} = \sum_{i \in N} u_i, \\ & x \in X. \end{aligned} \tag{12}$$

In the sequel, we fix $\lambda = 1.2$, that is, we give a higher weight to fairness. The ideal mechanism $\mathcal{D}^{ideal}$ enables us to compute the ideal values for different values of n and K. Recall that these ideal values represent the best values the principal can hope for, had shecomplete information about players' preferences. These are presented in Table 1.

[3] Note that the simplified setting in this section is similar to the piecewise constant valuations of Chen et al. (2013), but the pieces here are pre-determined rather than determined by the agents.

Table 1: Ideal values.

$n \setminus \lvert K\rvert$	2	3
2	1.27	1.35
5	1.35	1.56
20	1.42	1.69

In Hoang et al. (submitted), it was shown numerically (and in some instances, analytically) that the ideal mechanism is not BIC.

Now, we can apply our algorithms to compute the revelation mechanism of the ideal mechanism. The Bayesian-Nash equilibria for the following computations have been obtained with 250 iterations of Algorithm 1 when $n = 2$, 100 for $n = 5$ and 25 for $n = 20$. Also in Hoang et al. (submitted), it was experimentally shown that these numbers are sufficient to stabilize the sequence of best-reply return functions. Plus, we proceeded to 2,000 iterations to compute the objective value with a 90% confidence interval of $\pm.01$. Results are displayed in Table 2.

Table 2: Values of the revelation of the ideal mechanism.

$n \setminus \lvert K\rvert$	2	3
2	0.60	0.97
5	0.68	0.84
20	0.91	0.90

Disturbingly, the results show that the revelation of the ideal mechanism is much worse than the exact division. This means that, if the principal chooses to implement the ideal mechanism, then, eventually, agents will be revealing preferences in such a way that the division is much less optimal than it would be if we had just chosen an exact division in the first place.

Note that we have no theoretical guarantee of the existence nor the uniqueness of the Bayesian-Nash equilibrium of the ideal mechanism. However, the computational results in Hoang et al. (submitted) show that our algorithm does seem to always converge to the same return function profile, which suggests the existence and uniqueness of the Bayesian-Nash equilibrium. The same remark holds for other mechanisms we shall introduce.

3.3 Parameterized mechanisms

Let us now apply our approach to Bayesian mechanism design to determine an efficient and fair BIC mechanism. Interestingly, computations by Hoang et al. (submitted) show that the Bayesian-Nash equilibrium strategy of the ideal mechanism consists in overbidding undesirable attributes while underbidding desirable ones. Let us highlight shortly that, although this observation has been made for our specific ideal mechanism, it does unveil the fact that attempts to guarantee fairness lead to such bidding strategies on players' behalves.

To overrule this possibility, here, we force the mechanism to satisfy the following equality:

$$\forall i \in N, \forall k, l, \frac{x_{ik}}{x_{il}} = \left(\frac{a_{ik}}{a_{il}}\right)^p, \tag{13}$$

where p is some nonnegative number and a_i is the action of player i. This means that a player can have much more of attribute k than of attribute l only if shesays shereally prefers k to l. However, adding all equalities 13 to linear program 12 may lead to an infeasible program. Therefore, we instead add penalties for violating equalities 13, which take the form

$$\gamma_i = \max_{k,l} |x_{ik}a_{il}^p - x_{il}a_{ik}^p|. \tag{14}$$

Also, we add penalty $\delta_i = |u_i - \bar{u}|$ to each player, where u is the average utility. Denoting by w_δ the weight of penalties δ_i and by w_γ that of penalties γ_i, we get the following linear program:

$$\begin{aligned}
\underset{x,u,\bar{u},\delta,\gamma}{\text{Maximize}} \quad & n\bar{u} - w_\delta \sum_{i\in N} \delta_i - w_\gamma \sum_{i\in N} \gamma_i, & \\
\text{subject to:} \quad & u_i = \sum_{k=1}^{K} a_{ik} x_{ik}, & \forall i \in N, \\
& n\bar{u} = \sum_{i\in N} u_i, & \\
& \delta_i = |u_i - \bar{u}|, & \forall i \in N, \\
& \gamma_i \geq x_{ik} a^p_{il} - x_{il} a^p_{ik}, & \forall i \in N, \forall k, l \in K, \\
& x \in X. &
\end{aligned} \tag{15}$$

The parameterized mechanism $\mathcal{D}^t$ for $t = (p, w_\delta, w_\gamma)$ is defined as the mechanism having as its input an action profile a, and as its output a solution to linear program 15.

3.4 Computational results

To compute Bayesian-Nash equilibria s^t of parameterized mechanisms $\mathcal{D}^t$, we have used the same number of iterations as for the ideal mechanism, namely, 250 iterations of Algorithm 1 for $n = 2$, 100 for $n = 5$ and 25 for $n = 20$. Once the equilibria s^t were computed, we performed 2,000 iterations to compute the principal's objective value of the revelation mechanisms $\mathcal{D}^t_{Rev} = \mathcal{D}^t \circ s^t$. We obtained a 90% confidence interval of $\pm.01$.

Note that there is another error due to the approximation of the Bayesian-Nash equilibrium, which is harder to evaluate. To have a rough approximation of it, we have repeated the computation of the value of the revelation of the parameterized mechanism for $n = 5, |K| = 3, p = 0.5, w_\delta = 2$ and $w_\gamma = 10$. The results show a standard deviation of this computation of 0.01. This value is corroborated by the regularity of values of the tables, where variations between neighbor cells hardly exceed 0.03. However, our considerations here do not consider the bias due to the Bayesian-Nash equilibrium estimation.

Results are presented in the following Tables 3 to 8.

Table 3: Values of the revelations of parameterized mechanisms for $n = 2$ and $|K| = 2$.

$w = \dots$	$(1,1)$	$(1,10)$	$(1,100)$	$(2,1)$	$(2,10)$	$(2,100)$	$(5,1)$	$(2,10)$	$(5,100)$
$p = 0.3$	1.06	1.00	1.07	**1.12**	1.10	1.09	**1.12**	**1.11**	**1.13**
$p = 0.5$	**1.12**	**1.12**	**1.12**	**1.13**	**1.12**	**1.12**	1.10	1.07	1.10
$p = 0.7$	**1.11**	1.09	**1.13**	**1.12**	1.07	1.06	1.06	**1.11**	1.08

Table 4: Values of the revelations of parameterized mechanisms for $n = 2$ and $|K| = 3$.

$w = \dots$	$(1,1)$	$(1,10)$	$(1,100)$	$(2,1)$	$(2,10)$	$(2,100)$	$(5,1)$	$(2,10)$	$(5,100)$
$p = 0.3$	1.11	1.09	1.03	1.10	1.12	1.11	**1.17**	**1.16**	**1.16**
$p = 0.5$	1.05	1.13	1.10	**1.15**	1.14	**1.15**	**1.15**	1.14	1.14
$p = 0.7$	1.14	1.13	1.13	1.13	1.14	1.11	1.13	1.13	1.11

Table 5: Values of the revelations of parameterized mechanisms for $n = 5$ and $|K| = 2$.

$w = \dots$	$(1,1)$	$(1,10)$	$(1,100)$	$(2,1)$	$(2,10)$	$(2,100)$	$(5,1)$	$(2,10)$	$(5,100)$
$p = 0.3$	1.06	1.01	1.06	1.08	1.08	1.08	**1.16**	**1.16**	**1.16**
$p = 0.5$	1.10	1.09	1.11	**1.15**	**1.15**	1.14	1.10	1.10	1.09
$p = 0.7$	1.12	**1.15**	1.12	1.13	1.11	1.09	1.11	1.10	1.13

Table 6: Values of the revelations of parameterized mechanisms for $n = 5$ and $|K| = 3$.

$w = \ldots$	$(1,1)$	$(1,10)$	$(1,100)$	$(2,1)$	$(2,10)$	$(2,100)$	$(5,1)$	$(2,10)$	$(5,100)$
$p = 0.3$	1.08	1.05	1.07	1.17	1.15	1.15	**1.22**	**1.21**	**1.23**
$p = 0.5$	1.17	1.14	1.16	**1.22**	1.20	**1.21**	1.17	1.17	1.19
$p = 0.7$	1.20	1.20	**1.23**	**1.21**	1.20	**1.21**	1.15	1.16	1.15

Table 7: Values of the revelations of parameterized mechanisms for $n = 20$ and $|K| = 2$.

$w = \ldots$	$(1,1)$	$(1,10)$	$(1,100)$	$(2,1)$	$(2,10)$	$(2,100)$	$(5,1)$	$(2,10)$	$(5,100)$
$p = 0.3$	1.05	1.06	1.07	1.06	1.06	1.11	**1.14**	**1.13**	**1.14**
$p = 0.5$	1.09	1.11	1.08	1.09	**1.15**	**1.15**	1.09	1.08	1.11
$p = 0.7$	**1.15**	**1.14**	**1.13**	1.10	1.11	**1.14**	1.10	1.12	1.10

Table 8: Values of the revelations of parameterized mechanisms for $n = 20$ and $|K| = 3$.

$w = \ldots$	$(1,1)$	$(1,10)$	$(1,100)$	$(2,1)$	$(2,10)$	$(2,100)$	$(5,1)$	$(2,10)$	$(5,100)$
$p = 0.3$	1.22	**1.25**	**1.24**	1.19	1.20	1.18	1.20	**1.25**	**1.26**
$p = 0.5$	**1.25**	**1.24**	1.19	1.21	**1.24**	**1.24**	1.17	1.19	1.21
$p = 0.7$	**1.26**	1.21	**1.25**	**1.25**	1.23	**1.25**	1.18	1.17	1.19

Interestingly, there seems to be some pattern emerging in the tables: the best values for the parameters $t = (p, w_\delta, w_\gamma)$ seem to show up along the main diagonal. This signals an intricate implicit relationship between the parameters of our mechanisms. Indeed, increasing the value of p is in fact more or less equivalent to increasing the importance of equalizing the players' allocations. More precisely, the effect of an increase of p in the objective function of the heuristic mechanism is similar to the effect of an increase of the weights w_δ and w_γ. As a result, the mechanisms $\mathcal{D}^t$ and $\mathcal{D}^{t'}$ with $t = (p + \epsilon, w_\delta, w_\gamma)$ and $t' = (p, w_\delta + \epsilon, w_\gamma)$ are similar. This is why, we argue, their principal's objective values for the revelation of these mechanisms are close. The opposite also holds: if we decrease p, then it becomes important to increase the unfairness penalty, which can be done by increasing w_δ and w_γ.

In Table 9, we report the best BIC values found for revelations of parameterized mechanisms. These results show that there is a gain of at least 13% compared to the exact division. What is even more interesting, this gain seems to increase with the number of players and the number of attributes. This is not a big surprise as the greater diversity of preferences that comes with larger numbers of players and attributes offers more opportunities for improvement with respect to an exact division.

Table 9: Best values of revelations of parameterized mechanisms.

$n \setminus |K|$	2	3
2	1.13	1.17
5	1.16	1.23
20	1.15	1.26

Once again, we stress the generality of the results despite corresponding to our specific cake-cutting problem. Indeed, our family of parameterized mechanisms illustrates the relevancy of sacrificing the efficiency of the outcome to enforce greater incentive-compatibility.

4 Concluding remarks

In this paper, we have introduced a new approach to Bayesian mechanism design. The cornerstone of this approach is the computation of Bayesian-Nash equilibria with a fictitious play on return-function profiles, as presented by Hoang et al. (submitted). Indeed, using this computation, we can explore a space of BIC mechanisms, by tracking revelation mechanisms of a parameterized set of mechanisms. The application of this method to cake-cutting has yielded insightful results. We have shown that intuitively good mechanisms like the ideal mechanism turn out to be in fact quite bad when players play a Bayesian-Nash equilibrium. What is more, we have also constructed revelation mechanisms which represent significant improvements from more direct approaches. Crucially, the generality of our approaches mean that they can easily be extended to more practical cases of mechanism design.

References

X. Bei and Z. Huang. Bayesian incentive compatibility via fractional assignments. In Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, 720–733, 2011.

X. Bei, N. Chen, X. Hua, B. Tao, and E. Yang. Optimal proportional cake cutting with connected pieces. In Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI-12), 1263–1269, 2012.

S.J. Brams and A.D. Taylor. Fair Division: From cake-cutting to dispute resolution. Cambridge University Press, 1996.

I. Caragiannis, J.K. Lai, and A.D. Procaccia. Towards more expressive cake cutting. In IJCAI Proceedings-International Joint Conference on Artificial Intelligence, 22, 127, 2011.

Y. Chen, J.K. Lai, D.C. Parkes, and A.D. Procaccia. Truth, justice, and cake cutting. Games and Economic Behavior, 77(1), 284–297, 2013.

Y.J. Cohler, J.K. Lai, D.C. Parkes, and A.D. Procaccia. Optimal envy-free cake cutting. In Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11), 2011.

A. Gibbard. Manipulation of voting schemes: A general result. Econometrica: Journal of the Econometric Society, 41(4), 587–601, 1973.

J.D. Hartline. Mechanism design and approximation. Book draft. October, 2013.

J.D. Hartline and B. Lucier. Bayesian algorithmic mechanism design. In Proceedings of the 42nd ACM Symposium on Theory of Computing, 301–310, 2010.

J.D. Hartline, R. Kleinberg, and A. Malekian. Bayesian incentive compatibility via matchings. In Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, 734–747, 2011.

L.N. Hoang, F. Soumis, and G. Zaccour. The return function: A new computable perspective on bayesian-nash equilibria. submitted.

E. Mossel and O. Tamuz. Truthful fair division. In Algorithmic Game Theory, Lecture Notes in Computer Science Volume 6386, 288–299, 2010.

R.B. Myerson. Incentive compatibility and the bargaining problem. Econometrica: Journal of the Econometric Society, 47(1), 61–74, 1979.

R.B. Myerson. Optimal auction design. Mathematics of operations research, 6(1), 58–73, 1981.

D.C. Parkes. When analysis fails: Heuristic mechanism design via self-correcting procedures. In SOFSEM 2009: Theory and Practice of Computer Science, Lecture Notes in Computer Science Volume 5404, 62–66, 2009.

J. Robertson and W. Webb. Cake-cutting algorithms: Be fair if you can. CRC Press. 1998.

H. Steinhaus. The problem of fair division. Econometrica, 16, 101–104, 1948.