

Computational Aspects of Random Serial Dictatorship

HARIS AZIZ

NICTA and UNSW

and

FELIX BRANDT

Technische Universität München

and

MARKUS BRILL

Duke University

and

JULIÁN MESTRE

University of Sydney

Two fundamental problems in economics are voting and assignment. In both settings, *random serial dictatorship* is a well-established mechanism that satisfies anonymity, *ex post* efficiency, and strategyproofness. We present an overview of recent results on the computational complexity of problems related to random serial dictatorship.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms: Theory, Algorithms, Economics

Additional Key Words and Phrases: Game Theory, Solutions Concepts, Pareto Optimality, Computational Complexity

1. INTRODUCTION

Two fundamental problems in economics are voting and assignment. In the voting setting, *agents* express preferences over *alternatives* and a *social decision scheme* returns a probability distribution over the alternatives based on the agents' preferences [Gibbard, 1977]. In the assignment setting, *agents* express preferences over objects, usually called *houses* because only one object is assigned to each agent, and a *random assignment rule* returns a random assignment of the houses specifying the probability with which each house is allocated to each agent [Bogomolnaia and Moulin, 2001, Budish et al., 2013]. In both settings, randomization is crucial to achieve minimal fairness requirements such as anonymity and neutrality.

The mechanism known as *random serial dictatorship* (*RSD*) gives rise to both a desirable social decision scheme [Gibbard, 1977, Aziz et al., 2013b] and a desirable random assignment rule [Bogomolnaia and Moulin, 2001, Crès and Moulin, 2001].

Authors' addresses: haris.aziz@nicta.com.au, brandtf@in.tum.de, brill@cs.duke.edu, julian.mestre@sydney.edu.au.

In the voting setting, *RSD* selects a permutation of the agents uniformly at random and then chooses an alternative by sequentially allowing agents in the permutation to refine the set of feasible alternatives to their most preferred of the remaining alternatives. In the assignment setting, *RSD* selects a permutation of the agents uniformly at random and then lets one agent after another pick his most preferred of the remaining objects. In both settings, *RSD* is well-known to satisfy anonymity, strategyproofness, and *ex post* efficiency (i.e., it randomizes over Pareto optimal alternatives/allocations). In fact, it has been conjectured to be the only rule that satisfies these properties [see e.g., Lee and Sethuraman, 2011].¹

This paper surveys recent computational results regarding the probability of choosing an alternative in the context of voting and the probability of an agent getting a house. There are various reasons why computing the actual probabilities (rather than simply executing the mechanism) is important. In the assignment setting, the resulting probabilities of *RSD* can be viewed as fractional resource allocations such as in scheduling or other applications [see e.g., Svensson, 1994, Abdulkadiroğlu and Sönmez, 1998, Crès and Moulin, 2001]. Similarly, in voting, the probabilities returned by *RSD* can be interpreted as fractions of time or other resources allotted to the alternatives. Saban and Sethuraman [2013] mentioned identifying the conditions under which the *RSD* probabilities can be computed in polynomial time as an open problem. In another paper, Mennle and Seuken [2013] propose random hybrid assignment mechanisms that hinge on the *RSD* probabilities. Finally, the *RSD* probabilities have also been used in the design of a recent algorithm for cake cutting [Aziz and Ye, 2014].

2. PRELIMINARIES

A *voting problem* consists of a set $N = \{1, \dots, n\}$ of *agents* having preferences over a finite set A of *alternatives* where $|A| = m$. The preferences of agents over the alternatives are given by a preference profile $\succsim = (\succsim_1, \dots, \succsim_n)$ where, for each agent $i \in N$, $\succsim_i$ is a complete and transitive preference relation over A where $\succ_i$ denotes the strict part of the relation and $\sim_i$ denoted the indifference part. A preference relation $\succsim_i$ is *linear* if $a \succ_i b$ or $b \succ_i a$ for all distinct alternatives $a, b \in A$. For convenience, we will represent preference relations as comma-separated lists in which sets denote indifference classes.

The *serial dictatorship* rule is defined with respect to a permutation π over N . It starts with the set of all alternatives and then each agent in π successively refines the set of alternatives to the set of most preferred alternatives from the remaining set. *RSD* returns the serial dictatorship outcome with respect to a permutation that is chosen uniformly at random. If the outcome is not unique, we take the uniform probability distribution over the set of selected alternatives to enforce neutrality.

EXAMPLE 1 (ILLUSTRATION OF *RSD* IN VOTING). *Consider the following preference profile.*

$$\succsim_1: \quad \{a, b, c\}, d \qquad \succsim_2: \quad \{b, d\}, a, c \qquad \succsim_3: \quad c, \{a, b, d\}$$

¹When preferences are linear, Gibbard [1977] has shown that random dictatorship is the only strategyproof and *ex post* efficient social decision scheme.

For permutation 123, the serial dictatorship outcome is $\{b\}$. The RSD lottery is $[a : 0, b : 1/2, c : 1/2, d : 0]$.

An *assignment problem* is a triple $(N, H, \succsim)$, where N is a set of n agents, H is a set of m houses, and $\succsim = (\succsim_1, \dots, \succsim_n)$ is a preference profile that contains, for each agent i , a linear preference relation on the set of houses. Every random assignment specifies, for every agent i and every house h , the probability p_{ih} that house h is assigned to agent i . By the Birkhoff-von Neumann theorem, any such assignment can be obtained by a probability distribution over deterministic assignments. RSD chooses a permutation uniformly at random and then lets the agents sequentially take their most preferred object that has not yet been allocated according to the order of the permutation. It is easily observed that the assignment problem is a special case of the voting problem where the set of alternatives A is the set of all discrete assignments and the preferences of agents over A are induced by their preferences over H . Although agents have linear preferences over the houses, they are indifferent among different assignments in which they are allocated the same house.

EXAMPLE 2 (ILLUSTRATION OF RSD IN ASSIGNMENT). Consider the following preference profile.

$$\succsim_1: \quad a, b, c \qquad \qquad \succsim_2: \quad a, b, c \qquad \qquad \succsim_3: \quad b, a, c$$

Agents 1 and 2 are of the same type since they have identical preferences. For each permutation, one can compute the serial dictatorship outcome. For example, the permutation 123 yields the assignment $\{\{1, a\}, \{2, b\}, \{3, c\}\}$. The RSD random assignment is given by the following table.

	a	b	c
1	1/2	1/6	1/3
2	1/2	1/6	1/3
3	0	2/3	1/3

Although the assignment problem is a special case of the voting problem, this does not imply that positive algorithmic results for voting also hold within the assignment domain. The reason is that the translation of an assignment problem to a corresponding voting problem leads to an exponential blowup in the number of alternatives.

3. RESULTS

Recently, Aziz et al. [2013a] have shown that computing the RSD probabilities is #P-complete both in the voting and in the assignment setting. Independently, Saban and Sethuraman [2013] have shown the same result for the assignment setting.

THEOREM 1. (Aziz et al. [2013a]) In the voting setting, computing the RSD probabilities is #P-complete.

THEOREM 2. (Aziz et al. [2013a], Saban and Sethuraman [2013]) In the assignment setting, computing the RSD probabilities is #P-complete.

As a corollary, checking whether a given *RSD* probability is greater than or equal to some fixed $q \in (0, 1)$ is NP-hard. The reason is that since an *RSD* probability takes at most $n! + 1$ values, a polynomial-time algorithm for the problem can be used along with binary search to compute the exact probability in polynomial time. In the assignment setting, even checking whether an agent obtains a house with positive probability is intractable.

THEOREM 3. (*Saban and Sethuraman [2013]*) *In the assignment setting, checking whether an agent gets a house with positive RSD probability is NP-complete.*

A corollary of Theorem 3 is that the problem of computing *RSD* probabilities does not admit an FPRAS (fully polynomial-time randomized approximation scheme) unless the complexity class NP is equal to the complexity class RP [Saban and Sethuraman, 2013].

In contrast to the result in the assignment setting, Aziz et al. [2013a] showed that in the voting setting, the support of the *RSD* lottery can be computed in polynomial time. The algorithm greedily builds up a permutation, if possible, in which an agent is added to the partial permutation as long as the agent still keeps the target alternative in contention for being selected.

THEOREM 4. (*Aziz et al. [2013a]*) *In the voting setting, there is a polynomial-time algorithm to check whether an alternative receives positive RSD probability.*

The negative complexity results by Saban and Sethuraman [2013] and Aziz et al. [2013a] prompted the need to identify conditions under which *RSD* probabilities can be computed efficiently. A problem with parameter k belongs to the class FPT, and is said to be *fixed-parameter tractable*, if there exists an algorithm that solves every instance I of the problem in time $f(k) \cdot \text{poly}(|I|)$, where f is some computable function independent of I and poly is a polynomial. Aziz and Mestre [2014] examined the parameterized complexity of the problems with respect to the following parameters: the number of agent types, the number of alternatives, the number of alternative types, and the number of houses. Agents have the same type if they have identical preferences. Alternatives have the same type if each agent is indifferent among all of them.

THEOREM 5. (*Aziz and Mestre [2014]*) *In the voting setting,*

- (1) *There is an FPT algorithm for computing the RSD probabilities with parameter $T = \#$ of agent types.*
- (2) *There is an FPT algorithm for computing the RSD probabilities with parameter $q = \#$ of alternative types.*

THEOREM 6. (*Aziz and Mestre [2014]*) *In the assignment setting,*

- (1) *There is an FPT algorithm for computing the RSD probabilities with parameter $m = \#$ of houses.*
- (2) *There is a polynomial-time algorithm for computing the RSD probabilities if the number T of agent types is bounded.*

4. OPEN PROBLEMS

The recent work on computational aspects of random serial dictatorship gives rise to some interesting problems. In the assignment setting, Saban and Sethuraman [2013] proved that computing *RSD* probabilities does not admit an FPRAS unless $\text{NP}=\text{RP}$. For voting, it remains open whether there exists an FPRAS.

In the assignment setting, Aziz and Mestre [2014] proved that there is a polynomial-time algorithm if the number of agent types is bounded. It remains open whether there exist an FPT algorithm for computing the *RSD* probabilities for parameter $\#$ of agent types.

Finally, identifying fast exponential-time algorithms for computing the *RSD* probabilities in both the assignment and the voting setting is another direction for future work.

Acknowledgments. This material is based on work supported by Deutsche Forschungsgemeinschaft under grant BR 2312/7-2 and by a Feodor Lynen research fellowship of the Alexander von Humboldt Foundation. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

REFERENCES

- A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–702, 1998.
- H. Aziz and J. Mestre. Parametrized algorithms for random serial dictatorship. *Mathematical Social Sciences*, 2014.
- H. Aziz and C. Ye. Cake cutting algorithms for piecewise constant and piecewise uniform valuations. In *Proc. of 10th WINE*, 2014.
- H. Aziz, F. Brandt, and M. Brill. The computational complexity of random serial dictatorship. *Economics Letters*, 121(3):341–345, 2013a.
- H. Aziz, F. Brandt, and M. Brill. On the tradeoff between economic efficiency and strategyproofness in randomized social choice. In *Proc. of 12th AAMAS Conference*, pages 455–462. IFAAMAS, 2013b.
- A. Bogomolnaia and H. Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100(2):295–328, 2001.
- E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom. Designing random allocation mechanisms: Theory and applications. *American Economic Review*, 103(2):585–623, 2013.
- H. Crès and H. Moulin. Scheduling with opting out: Improving upon random priority. *Operations Research*, 49(4):565–577, 2001.
- A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3):665–681, 1977.
- T. Lee and J. Sethuraman. Equivalence results in the allocation of indivisible objects: a unified view. Working paper, August 2011.
- T. Mennle and S. Seuken. Partially strategyproof mechanisms for the assignment problem. Technical Report arXiv:1303.2558, arXiv.org, 2013.
- D. Saban and J. Sethuraman. The complexity of computing the random priority allocation matrix. In *Proc. of 9th WINE*, LNCS, page 421, 2013.
- L.-G. Svensson. Queue allocation of indivisible goods. *Social Choice and Welfare*, 11:323–330, 1994.