A Game-Theoretic View of Randomized Fair Multi-Agent Optimization

Hugo Gilbert, Olivier Spanjaard

A Game-Theoretic View of Randomized Fair Multi-Agent Optimization

Hugo Gilbert, Olivier Spanjaard

Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, Paris, France
{hugo.gilbert,olivier.spanjaard}@lip6.fr

Abstract. We tackle fair multi-agent optimization problems and use a generalized Gini index to determine a fair and efficient solution. We claim that considering mixed solutions (i.e., lotteries over solutions) enables to enhance the fairness of an optimal solution. Interpreting a fair multi-agent optimization problem as a zero-sum two-player game between an optimization player choosing a solution and an adversary which has some control over the payoffs of the game, we propose two methods (a cutting-plane method and a double oracle method) to compute an optimal mixed solution. Numerical tests are provided to compare their efficiency.

1 Introduction

Multiagent optimization deals with problems where multiple agents are involved in the choice of a feasible solution. Multiagent optimization procedures are required in many problems, such as proportional representation (typically winner determination under the Chamberlin-Courant multiwinner voting rules, which determines a set of representatives minimizing the total dissatisfaction of the voters [22]), group recommendation (e.g., movies to put on a plane's entertainment system [21]), fair division of indivisible goods [14], or paper assignment problems (assigning reviewers to conference paper submissions [10]).

We are more especially interested here in *fair* multiagent optimization, i.e. in procedures favoring solutions that fairly share satisfaction among agents. There are several ways of formalizing "fairness". We mean here by fair optimization that, when considering the vector of agent's satisfactions, it should be both Pareto optimal (i.e., the satisfaction of an agent cannot be improved without decreasing that of another agent) and well-balanced (in a sense to be formalized later). Nevertheless, due to conflicting agents' preferences, all feasible solutions can be unfair to some extent. In this concern, randomness can help to "even things out", by determining a probability distribution over feasible solutions (called a *mixed* solution in the game theoretic terminology) instead of looking for a fair deterministic solution (a fair *pure* solution) that might not exist.

Example 1 (Machina's mom [16]) *A mother with two children has one indivisible treat. She can give it to either one of her children but not both. It is reasonable to imagine that the mother would prefer tossing a coin to decide which* *child should have the treat instead of choosing herself (i.e., choosing between the two feasible pure assignments where a child receives the treat while the other has nothing), so that both children have equal chances of having the treat.*

Randomization has mainly been studied for the assignment problem where, given n objects and n agents with individual preferences on the objects, one wishes to assign exactly one object to each agent in a fair and efficient manner. An extension of mom's approach to this problem with n agents is called the random serial dictatorship procedure: order the agents uniformly at random, and let them successively choose an object in that order [1]. This procedure induces a *bistochastic* matrix $P = (p_{ij})$ where p_{ij} is the probability that agent i obtains object j. By the Birkhoff-von Neumann theorem, matrix P may be represented by a probability distribution over pure assignments (i.e., a mixed assignment) whose inferred allocation probabilities coincide with p_{ij}. Note that other sophisticated random assignment rules can be found in the literature [4, 12]. These approaches are however hard to transpose to other multiagent optimization problems.

The notion of *popular mixed assignment* can nevertheless be transposed to any multiagent optimization problem. A mixed assignment p is popular if there is no assignment q such that the expected number of agents who prefer the outcome of q to that of p is greater than $n/2$ [13]. By the minimax theorem, a popular mixed assignment always exists, and Kavitha *et al.* proposed a linear programming approach to compute it in polynomial time. Regarding fairness, Aziz *et al.* [2] have shown that there always exists a popular mixed assignment that satisfies *equal treatment of equals*, i.e. agents with identical preferences receive identical random allocations. Equal treatment of equals is considered as a "minimal test for fairness" [19].

Yet, the choice of a popular mixed solution does not preclude the possibility that a minority of agents are unsatisfied with all pure solutions considered in that mixed solution. In this paper, following Lesca and Perny [15] and Endriss [8] who also studied fair optimization but in a deterministic setting, we adopt the viewpoint of the measurement of inequality [19] to compare vectors of agents' expected utilities induced from mixed solutions. The Pigou-Dalton principle is the basic postulate of inequality measurement. It states that transferring some utility from one agent to another so as to reduce the difference in their welfare should not reduce social welfare [19]. Our aim is to provide a procedure to determine a mixed solution optimizing a criterion compatible with this principle. In this concern, the *generalized Gini inequality indices* (a subclass of *Ordered Weighted Averages*, abbreviated by OWA) are well-known to satisfy a number of appealing fairness properties, among which the Pigou-Dalton principle [23]. In the following, we provide two generic procedures for computing a mixed solution optimizing a generalized Gini inequality index. They are both based on a game-theoretic view of fair multiagent optimization problems. One procedure is a cutting plane method [7] while the other is a double oracle algorithm [18].

The paper is organized as follows. Section 2 presents the randomized fair optimization problem. A game-theoretic view is given in Section 3, which allows us to derive two solution methods. Numerical tests are presented in Section 4.

2 Fair Optimization with OWA

We adopt the same setting as in previous works dealing with inequality measurement for fair multiagent optimization [8, 15]. Let $\mathcal{P}$ be a multiagent optimization problem, and $\mathcal{N} = \{1, \ldots, n\}$ the finite set of agents involved in $\mathcal{P}$. We denote by $\mathcal{X} \subseteq \{0,1\}^p$ the set of feasible solutions of $\mathcal{P}$ and we assume that $\mathcal{X}=\{\mathbf{x} \in \{0,1\}^p : A\mathbf{x} = \mathbf{b}\}$[1] where A is called *constraint matrix* and $\mathbf{b}$ is integral. A feasible solution of $\mathcal{P}$ is thus represented by a binary vector $\mathbf{x} \in \mathcal{X}$. Each agent i is endowed with a utility function $u_i : \mathcal{X} \to \mathbb{R}^+$ (to maximize, w.l.o.g.) defined by $u_i(\mathbf{x}) = \sum_{j=1}^{p} u_{ij}x_j$ where u_{ij} is the utility of having $x_j = 1$ for agent i. Every solution $\mathbf{x}$ induces therefore a utility vector $\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), \ldots, u_n(\mathbf{x}))$. To compare two vectors, one can use an *aggregation criterion* $F : (\mathbb{R}^+)^n \to \mathbb{R}$ such that a vector $\mathbf{x}$ is weakly preferred to $\mathbf{y}$ if $F(\mathbf{x}) \geq F(\mathbf{y})$.

2.1 The OWA Operator

We recall that the aim of fair multiagent optimization is to determine a solution that is both efficient and fair. The concept of efficiency is captured by the notion of Pareto-optimality:

Definition 1 *A vector* $\mathbf{y}$ Pareto-dominates *a vector* $\mathbf{y}'$ *if:*

i) $\forall i \in \{1, \ldots, n\}, y_i \geq y'_i$ *ii)* $\exists i \in \{1, \ldots, n\}, y_i > y'_i$

A solution $\mathbf{x}$ *is said to be* Pareto-optimal *in a set* S *of solutions iff there is no* $\mathbf{x}' \in S$ *such that* $\mathbf{u}(\mathbf{x}')$ *Pareto-dominates* $\mathbf{u}(\mathbf{x})$.

An easy way to obtain a Pareto-optimal solution is to maximize a Weighted Average (WA) of utilities:

Definition 2 *Let* $\mathbf{w} = (w_1, w_2, \ldots w_n)$ *be a vector of weights. The* $\texttt{WA}_{\mathbf{w}}(\cdot)$ *operator induced by* $\mathbf{w}$ *is defined by:* $\forall \mathbf{x} \in \mathcal{X}, \texttt{WA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x})) = \sum_{i=1}^{n} w_i u_i(\mathbf{x})$.

Optimizing a WA can often be performed very efficiently as it reduces the multiagent problem to a standard single criterion problem. However, such criterion may lead to an unfair solution as this criterion compensates between the utility scores of the different agents. For instance, for two agents and $\mathbf{w} = (1, 1)$, the utility vector $(1, 10)$ is preferred to $(5, 5)$ for the $\texttt{WA}_{\mathbf{w}}$ criterion. A natural condition to model fairness is the Pigou-Dalton transfer principle [19]:

Definition 3 *A criterion* F *satisfies the* transfer principle *if:* $F(u_1, \ldots, u_i - \epsilon, \ldots, u_j + \epsilon, \ldots, u_n) > F(u_1, \ldots, u_n)$ *for* $u_j - u_i < \epsilon < u_i - u_j$*, for any* $\mathbf{u} \in (\mathbb{R}^+)^n$.

This condition states that the overall satisfaction should be improved by any transfer of utility from a "richer" agent to a "poorer" one.

A well known criterion satisfying the transfer principle and whose optimization yields a Pareto-optimal solution is the generalized Gini inequality index [3]. This criterion is also known in multicriteria decision making under the name of *Ordered Weighted Average* (OWA) [24].

[1] Note that we adopt the convention to use bold letters to denote vectors.

Definition 4 *Let* $\mathbf{w} = (w_1, \ldots, w_n)$ *be a vector of weights. The* $\mathtt{OWA}_{\mathbf{w}}(\cdot)$ *operator induced by* $\mathbf{w}$ *is defined by:* $\forall \mathbf{x} \in \mathcal{X}, \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x})) = \sum_{i=1}^{n} w_i u_{\sigma(i)}(\mathbf{x})$*, where* σ *is the permutation of* $\{1, \ldots, n\}$ *such that* $u_{\sigma(1)}(\mathbf{x}) \leq u_{\sigma(2)}(\mathbf{x}) \leq \ldots \leq u_{\sigma(n)}(\mathbf{x})$.

Importantly, note that the set of generalized Gini inequality indices is the subclass of OWA operators where $w_i > w_{i+1}$ for $i \in \{1, \ldots, n-1\}$. This property insures that the transfer principle holds. From now on, we restrict our attention to this subclass of OWAs and weights w_i's are therefore assumed to be decreasing.

OWA operators are very general operators and encompass the average, the minimum and the maximum operators. Moreover, when $w_i \gg w_{i+1}$ for all i, the OWA operator behaves as the leximin operator [19].

We now recall the definition and some properties of *Lorenz vectors*, as thinking about OWA as an aggregation over the components of a Lorenz vector helps provide further insights into the use of this operator for inequality measurement.

Definition 5 *Given a vector* $\mathbf{y} = (y_1, \ldots, y_n)$*, the* Lorenz vector *of* $\mathbf{y}$ *is defined by* $\mathcal{L}(\mathbf{y}) = (l_1(\mathbf{y}), \ldots, l_n(\mathbf{y}))$*, where* $l_i(\mathbf{y})$ *is the sum of the* i *smallest elements of* $\mathbf{y}$*. More formally, let* σ *be the permutation of* $\{1, \ldots, n\}$ *such that* $y_{\sigma(1)} \leq y_{\sigma(2)} \leq \ldots \leq y_{\sigma(n)}$*, then* $l_i(\mathbf{y}) = \sum_{j=1}^{i} y_{\sigma(j)}$.

A vector $\mathbf{y}$ is said to *Lorenz-dominate* a vector $\mathbf{y}'$ if $\mathcal{L}(\mathbf{y})$ Pareto-dominates $\mathcal{L}(\mathbf{y}')$. In the bi-dimensional Cartesian coordinate system, the *Lorenz curve* associated with a vector $\mathbf{y}$ is the piecewise-linear curve connecting the points $(i/n, l_i(\mathbf{y})/l_n(\mathbf{y}))$ for $i = 0$ to n, where $l_0(\mathbf{y}) = 0$. Graphically, a vector $\mathbf{y}$ Lorenz-dominates a vector $\mathbf{y}'$ iff the Lorenz curve of $\mathbf{y}$ is above the one of $\mathbf{y}'$. Lorenz dominance is a key concept in inequality measurement due to the following:

Theorem 1 *[5] For any pair of positive vectors* $\mathbf{y}, \mathbf{y}'$*, if* $\mathbf{y}$ *Pareto-dominates* $\mathbf{y}'$*, or if* $\mathbf{y}'$ *is obtained from* $\mathbf{y}$ *by a Pigou-Dalton transfer, then* $\mathbf{y}$ *Lorenz-dominates* $\mathbf{y}'$*. Conversely, if* $\mathbf{y}$ *Lorenz-dominates* $\mathbf{y}'$*, then there exists a sequence of Pigou-Dalton transfers and/or Pareto-improvements to transform* $\mathbf{y}'$ *into* $\mathbf{y}$.

In other words, given two solutions $\mathbf{x}$ and $\mathbf{x}'$, if $\mathbf{u}(\mathbf{x})$ Lorenz-dominates $\mathbf{u}(\mathbf{x}')$, then $\mathbf{x}$ should be preferred to $\mathbf{x}'$ from the viewpoint of efficiency and fairness.

Interestingly enough, an OWA operator can be rewritten as:

$$\forall \mathbf{x} \in \mathcal{X}, \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x})) = \sum_{i=1}^{n} \lambda_i l_i(\mathbf{u}(\mathbf{x}))$$

where $\lambda = (w_1 - w_2, w_2 - w_3, \ldots, w_{n-1} - w_n, w_n)$. Thus, provided $w_i > w_{i+1}$ for $i \in \{1, \ldots, n-1\}$, a solution optimizing an OWA operator is always Lorenz optimal (where Lorenz optimality is defined similarly to Pareto optimality).

The weights initially proposed for the Gini social-evaluation function are:

$$w_i = (2(n-i)+1)/n^2 \tag{1}$$

With these weights, $\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x}))$ has a nice graphical interpretation: indeed, value $1 - \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x}))/\mu$, where $\mu = \sum_{i=1}^{n} u_i(\mathbf{x})/n$, equals two times the area between the Lorenz curve of $\mathbf{x}$ and the diagonal representing the ideal distribution of utilities (where $u_i(\mathbf{x}) = \mu$ for every agent). Obviously, for a given μ, the narrower this area the better, which is consistent with the maximization of OWA.

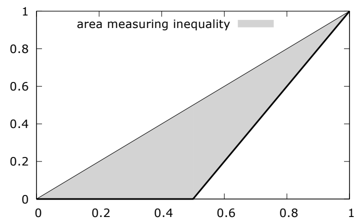


Fig. 1: Lorenz curve and inequality measurement.

Example 2 (Machina's mom cont') *Let us come back to Example 1. We have* $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^2 : x_1 + x_2 = 1\}$*, where* $x_i = 1$ *if child* i *receives the treat. There are therefore two pure solutions* $\mathbf{x^1} = (1,0)$ *and* $\mathbf{x^2} = (0,1)$*. We assume that* $u_i(\mathbf{x}) = x_i$ *for* $i \in \{1,2\}$ *(the utility of* i *is 1 if she receives the treat, 0 otherwise). We have* $\mathcal{L}(u(\mathbf{x}^1)) = \mathcal{L}(u(\mathbf{x}^2)) = (0,1)$*. The Lorenz curves associated with* $\mathbf{u}(\mathbf{x^1})$ *and* $\mathbf{u}(\mathbf{x^2})$ *are therefore identical. They correspond to the bold curve in Figure 1. The ideal distribution of utilities would be* $(0.5, 0.5)$*, yielding a Lorenz vector* $(0.5, 1)$*. The Lorenz curve associated with this ideal distribution is the diagonal in Figure 1. The surface of the area in light gray is an indicator of the level of fairness.*

2.2 The OWA Operator with Mixed Solutions

We now investigate the properties that hold for mixed solutions optimizing an OWA operator. A mixed solution is a lottery over solutions in $\mathcal{X}$ and is denoted by $P_{\mathcal{X}}$. The set of all possible mixed solutions is denoted by $\Delta_{\mathcal{X}}$. Given a solution $P_{\mathcal{X}}$, we denote by $P_{\mathcal{X}}(\mathbf{x})$ the probability assigned to pure solution $\mathbf{x}$. As usual, the definition of a utility vector is extended by linearity to mixed solutions. More formally, $u_i(P_{\mathcal{X}}) = \sum_{\mathbf{x} \in \mathcal{X}} P_{\mathcal{X}}(\mathbf{x}) u_i(\mathbf{x})$. Note that an OWA optimal mixed solution is always Pareto-optimal in $\Delta_{\mathcal{X}}$, by compatibility of OWA with Pareto dominance [9].

We now illustrate the enhanced abilities of OWA optimal mixed solutions w.r.t. fairness, compared to pure solutions.

Example 3 (Machina's mom cont') *Coming back again to Example 1, consider pure solutions* $\mathbf{x^1}, \mathbf{x^2}$ *and the mixed solution* $P_{\mathcal{X}}$ *defined by* $P_{\mathcal{X}}(\mathbf{x^i}) = 0.5$ *for* $i \in \{1,2\}$*. If the weights* w_i *are defined as in Equation 1, we have:*

$$\begin{aligned}
\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x^1})) &= \mathtt{OWA}_{\mathbf{w}}((1,0)) = 0.75 \cdot 0 + 0.25 \cdot 1 = 0.25 \\
\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(\mathbf{x^2})) &= \mathtt{OWA}_{\mathbf{w}}((0,1)) = 0.75 \cdot 0 + 0.25 \cdot 1 = 0.25 \\
\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(P_{\mathcal{X}})) &= \mathtt{OWA}_{\mathbf{w}}((0.5,0.5)) = 0.75 \cdot 0.5 + 0.25 \cdot 0.5 = 0.5
\end{aligned}$$

because $\mathbf{u}(P_{\mathcal{X}}) = 0.5\mathbf{u}(\mathbf{x^1}) + 0.5\mathbf{u}(\mathbf{x^2}) = (0.5, 0.5)$*. The mixed solution* $P_{\mathcal{X}}$ *is therefore preferred to both pure solutions. This is not surprising as the Lorenz curve associated with* $\mathbf{u}(P_{\mathcal{X}})$ *coincides with the diagonal.*

As witnessed by this example, for mixed solutions in allocation problems, the Pigou-Dalton principle guarantees the desirable property of equal treatment of equals (more precisely, there always exists *an* optimal mixed solution where agents with equal preferences receive the same random allocation), which is of course not the case in pure strategies.

In the next section, we turn to the question of computing an OWA optimal mixed solution.

3 A Game-Theoretic View

To model the problem of determining an OWA optimal mixed solution as the determination of a mixed Nash Equilibrium (NE) in a zero-sum two-player game, the following observation, *holding only for decreasing weights* w_i, reveals useful:

$$\forall P_{\mathcal{X}} \in \Delta_{\mathcal{X}}, \quad \mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(P_{\mathcal{X}})) = \min_{\sigma \in \Sigma} \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(P_{\mathcal{X}}))$$

where Σ is the set of all permutations of $\{1, \ldots, n\}$ and $\mathbf{w}_\sigma = (w_{\sigma(1)}, \ldots, w_{\sigma(n)})$. Thus, maximizing $\mathtt{OWA}_{\mathbf{w}}(\mathbf{u}(P_{\mathcal{X}}))$ amounts to the max min optimization problem:

$$\max_{P_{\mathcal{X}} \in \Delta_{\mathcal{X}}} \min_{\sigma \in \Sigma} \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(P_{\mathcal{X}})).$$

This is exactly the problem faced by player 1 in the zero-sum two-player game where the sets of pure strategies are the set of feasible solutions $\mathcal{X}$ for player 1 (also called x-player), the set of permutations Σ for player 2 (also called σ-player), and the payoffs are given by values $\mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(\mathbf{x}))$ for $\mathbf{x} \in \mathcal{X}$ and $\sigma \in \Sigma$. Similarly to the x-player, we denote by P_Σ a mixed strategy of the σ-player, and by Δ_Σ the set of all her mixed strategies. Note that the WA operator is extended to mixed strategies by linearity:

$$\mathtt{WA}_{\mathbf{w}_{\mathbf{P}_\Sigma}}(\mathbf{u}(P_{\mathcal{X}})) = \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\sigma \in \Sigma} P_{\mathcal{X}}(\mathbf{x}) P_\Sigma(\sigma) \mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(\mathbf{x})).$$

Given any two-player zero-sum game, determining a mixed NE can be done by linear programming [6]. Indeed, assume that player 1 (resp. player 2) has k (resp. m) pure strategies and that A_{ij} denotes the payoff of player 1 when strategies i and j are played by respectively player 1 and player 2. Then, a mixed NE of this game can be determined by the linear program $\mathcal{P}_{NE}$ (NE for Nash Equilibrium) given on the left below:

$$\mathcal{P}_{NE}: \begin{cases} \max\limits_{v, p_1, \ldots, p_k} v \\ \sum\limits_{i=1}^{k} p_i A_{ij} \geq v \quad \forall j \in \{1, \ldots, m\} & (2) \\ \sum\limits_{i=1}^{k} p_i = 1 \\ v \in \mathbb{R} \quad p_i \geq 0 \quad \forall i \in \{1, \ldots, k\} \end{cases}$$

$$\mathcal{P}_{FO}: \begin{cases} \max\limits_{v, p_{\mathbf{x}}: \mathbf{x} \in \mathcal{X}} v \\ \sum\limits_{i=1}^{n} w_{\sigma(i)} (\sum\limits_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}} u_i(\mathbf{x})) \geq v \quad \forall \sigma \in \Sigma & (3) \\ \sum\limits_{\mathbf{x} \in \mathcal{X}} p_{\mathbf{x}} = 1 \\ v \in \mathbb{R} \quad p_{\mathbf{x}} \geq 0 \quad \forall \mathbf{x} \in \mathcal{X} \end{cases}$$

where p_i denotes the probability that player 1 selects pure strategy i and v denotes the value of the game. In our setting, $\mathcal{P}_{NE}$ is rewritten as program $\mathcal{P}_{FO}$ (FO for Fair Optimization), given on the right above, where constraints 3 are just the specification of constraints 2 in $\mathcal{P}_{NE}$ to our setting. However, program $\mathcal{P}_{FO}$ is too large to be solved directly as it involves an exponential number of variables and constraints ($|\mathcal{X}| + 1$ variables and $|\Sigma| + 1$ constraints).

3.1 Cutting Plane Method

The method proposed here is based on a reformulation of $\mathcal{P}_{FO}$ that involves a polynomial number of variables, by redefining the solution space. We recall that a solution $\mathbf{x} \in \mathcal{X}$ is encoded by a vector of p binary variables. A mixed solution $P_{\mathcal{X}}$ thus induces a vector $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_p)$ where $\tilde{x}_i = \sum_{\mathbf{x} \in \mathcal{X}} P_{\mathcal{X}}(\mathbf{x}) x_i$. We denote by $\text{Conv}(\mathcal{X}) = \{\sum_{\mathbf{x} \in \mathcal{X}} P_{\mathcal{X}}(\mathbf{x})\mathbf{x} : P_{\mathcal{X}} \in \Delta_{\mathcal{X}}\}$ the convex hull of $\mathcal{X}$. Given $\tilde{\mathbf{x}} \in \text{Conv}(\mathcal{X})$, we say that $P_{\mathcal{X}}$ *implements* $\tilde{\mathbf{x}}$ if $\sum_{\mathbf{x} \in \mathcal{X}} P_{\mathcal{X}}(\mathbf{x})\mathbf{x} = \tilde{\mathbf{x}}$. By Carathéodory's theorem, for any $\tilde{\mathbf{x}} \in \text{Conv}(\mathcal{X})$ there exists a mixed solution that implements it with a combination of at most $p+1$ pure solutions.

Example 4 *Consider a one-to-one assignment problem with 3 agents. We denote by (i,j,k) the allocation where agents 1 (resp. 2, 3) receives object i (resp. j,k). The set Conv($\mathcal{X}$) includes here all bistochastic matrices of dimension 3. The bistochastic matrix*

$$\tilde{\mathbf{x}} = \begin{pmatrix} 0.8 & 0 & 0.2 \\ 0.2 & 0.3 & 0.5 \\ 0 & 0.7 & 0.3 \end{pmatrix}$$

can be implemented by the mixed solution $P_{\mathcal{X}}$ where pure assignments $(1,2,3)$, $(1,3,2)$ and $(2,3,1)$ are returned with probabilities 0.3, 0.5 and 0.2 respectively.

Hence, for any fair multi-agent optimization problem, problem $\mathcal{P}_{FO}$ can be rewritten as follows:

$$\widetilde{\mathcal{P}}_{FO} \begin{cases} \max\limits_{v, \tilde{x}_1, \ldots, \tilde{x}_p} v \\ v \leq \sum\limits_{i=1}^{n} w_{\sigma(i)} u_i(\tilde{\mathbf{x}}) \qquad \forall \sigma \in \Sigma & (4) \\ \tilde{\mathbf{x}} \in \text{Conv}(\mathcal{X}) & (5) \end{cases}$$

because $u_i(\tilde{\mathbf{x}}) = \sum_{\mathbf{x} \in \mathcal{X}} P_{\mathcal{X}}(\mathbf{x}) u_i(\mathbf{x})$ if $P_{\mathcal{X}}$ implements $\tilde{\mathbf{x}}$. Note that the linearization used here for the OWA objective function (valid only if the weights are decreasing) coincides with the one proposed by Ogryczak and Śliwiński [20]; by comparison, the above reasoning proves that determining an optimal mixed solution amounts to determining an optimal solution in Conv($\mathcal{X}$).

Even if the number of constraints in 4 (resp. in 5) is (resp. may be) exponential in n (resp. p) as $|\Sigma| = n!$ (resp. the number of facets of Conv($\mathcal{X}$) may be exponential in p), these constraints can be handled efficiently by resorting to a *cutting plane approach*. A cutting plane approach makes it possible to solve a linear program involving an exponential number of constraints to define a polyhedron P, provided there exists a *separation oracle*. In program $\widetilde{\mathcal{P}}_{FO}$, polyhedron P is defined by constraints 4 and 5. Given $(v, \tilde{\mathbf{x}}) \in \mathbb{R} \times [0,1]^p$, a separation oracle should determine whether $(v, \tilde{\mathbf{x}})$ belongs to P or not, and finds a separating hyperplane in the latter case. The separation oracle we propose consists of a separation oracle for the polyhedron P_4 defined by constraints in 4 and a separation oracle for the polyhedron P_5 defined by constraints in 5:

– Given $(v, \tilde{\mathbf{x}})$, a separation oracle for P_4 consists in sorting values $u_1(\tilde{\mathbf{x}}) \ldots u_n(\tilde{\mathbf{x}})$ to determine a permutation that minimizes $\sum_{i=1}^n w_{\sigma(i)} u_i(\tilde{\mathbf{x}})$. Sorting these values can of course be performed in polynomial time. If $v \leq \sum_{i=1}^n w_{\sigma(i)} u_i(\tilde{\mathbf{x}})$, then $\tilde{\mathbf{x}}$ belongs to P_4, otherwise $v = \sum_{i=1}^n w_{\sigma(i)} u_i(\tilde{\mathbf{x}})$ defines a separating hyperplane. It amounts to generating a most violating constraint in 4, if any.

– Regarding polyhedron P_5, if we can optimize over $\mathcal{X}$ in polynomial time, then we can separate over $\mathrm{Conv}(\mathcal{X})$ in polynomial time by the polynomial time equivalence of optimization and separation [11] (optimization $\rightarrow$ separation).

Combining both oracles yields a separation oracle for polyhedron $P = P_4 \cap P_5$, which is polynomial time if we can optimize over $\mathcal{X}$ in polynomial time. In this case, the complexity of solving $\widetilde{\mathcal{P}}_{FO}$ is polynomial by the polynomial time equivalence of optimization and separation (separation $\rightarrow$ optimization).

Once an OWA optimal solution $\tilde{\mathbf{x}} \in \mathrm{Conv}(\mathcal{X})$ has been found, it remains to *actually* compute a mixed solution in $\Delta_{\mathcal{X}}$ that implements $\tilde{\mathbf{x}}$. For this purpose, following Mastin et al. [17], we consider the linear program $\mathcal{P}_{\widetilde{\Delta}\rightarrow\Delta}$ below. Clearly, a feasible solution of $\mathcal{P}_{\widetilde{\Delta}\rightarrow\Delta}$ induces a mixed solution $P_{\mathcal{X}}$ that implements $\tilde{\mathbf{x}}$, by setting $P_{\mathcal{X}}(\mathbf{x}) = p_{\mathbf{x}}$. Nevertheless, this program has an exponential number of variables ($|\mathcal{X}|$ variables $p_{\mathbf{x}}$). To tackle this issue, we consider the dual program $\mathcal{D}_{\widetilde{\Delta}\rightarrow\Delta}$ below, where w is the dual variable of constraint 7 and variables w_i are the dual variables of constraints in 6.

$$
\begin{cases}
\min\limits_{p_{\mathbf{x}}:\mathbf{x}\in\mathcal{X}} 0 & \\
\sum\limits_{\mathbf{x}\in\mathcal{X}} p_{\mathbf{x}} x_i = \tilde{x}_i \quad \forall i \in \{1,\dots,p\} & (6)\\
\sum\limits_{\mathbf{x}\in\mathcal{X}} p_{\mathbf{x}} = 1 & (7)\\
p_{\mathbf{x}} \geq 0 \quad \forall \mathbf{x} \in \mathcal{X} &
\end{cases}
$$

$\mathcal{P}_{\widetilde{\Delta}\rightarrow\Delta}$

$$
\begin{cases}
\max\limits_{w,u_1,\dots,u_p} w - \sum\limits_{i=1}^{p} \tilde{x}_i w_i & \\
w \leq \sum\limits_{i=1}^{p} w_i x_i \quad \forall \mathbf{x} \in \mathcal{X} & (8)\\
w \in \mathbb{R} \quad w_i \in \mathbb{R} \quad \forall i \in \{1,\dots,p\} & (9)
\end{cases}
$$

$\mathcal{D}_{\widetilde{\Delta}\rightarrow\Delta}$

The probability of a pure solution $\mathbf{x}$ in the resulting mixed solution is given by the dual variable of the corresponding constraint in 8. The separation oracle for constraints in 8 requires to solve the single objective minimization variant of $\mathcal{P}$ where the cost of having $x_i = 1$ is given by variable w_i and where one aims at minimizing the sum of costs. If this problem is polynomial, the complexity of solving $\mathcal{D}_{\widetilde{\Delta}\rightarrow\Delta}$ is polynomial by the polynomial time equivalence of optimization and separation (separation $\rightarrow$ optimization), and the mixed solution resulting from the optimization is guaranteed to have a polynomially bounded support in $\mathcal{X}$ (the support is the set of pure solutions with nonzero probability). By solving successively $\widetilde{\mathcal{P}}_{FO}$ and $\mathcal{D}_{\widetilde{\Delta}\rightarrow\Delta}$, we conclude:

Theorem 2 *Any randomized fair optimization problem whose single objective optimization variant is polynomial in p is polynomially solvable in p and n.*

This result applies for instance to fair multiagent one-to-one and many-to-many allocation problems, and to fair multiagent matroid problems (where each agent assigns a cost to each element of the ground set).

Note that the polynomial time equivalence of optimization and separation is mainly a theoretical tool, since it is well known that the resulting algorithms do not appear to be efficient in practice [7]. For the numerical tests presented in Section 4, we therefore implemented a non-polynomial cutting plane method (but more efficient in practice) where constraints in 4 are iteratively generated and constraints in 5 are explicitly listed, which is harmless if the constraint matrix A is of polynomial size and is totally unimodular (TU). Indeed, if A is TU, $\mathrm{Conv}(\mathcal{X})$ is obtained by linear programming relaxation and thus the oracle for P_5 is unnecessary. In the next subsection, we present another algorithm to solve the game induced by the randomized OWA optimization problem. This algorithm has no polynomial time guarantee, but is competitive in practice even if matrix A is not TU.

3.2 Double Oracle Algorithm

The game can be solved by specifying a double oracle algorithm adapted to our problem. A double oracle algorithm finds an NE for a finite zero-sum two player game where a best response procedure (also called oracle) exists for each player. Given a mixed strategy $P_{\mathcal{X}}$ (resp. P_{Σ}), $BR_{\sigma}(P_{\mathcal{X}})$ (resp. $BR_{\mathbf{x}}(P_{\Sigma})$) returns a pure strategy σ (resp. $\mathbf{x}$) that minimizes $\mathtt{WA}_{\mathbf{w}_{\sigma}}(\mathbf{u}(P_{\mathcal{X}}))$ (resp. maximizes $\mathtt{WA}_{P_{\Sigma}}(\mathbf{u}(\mathbf{x}))$). The algorithm starts by considering only small subsets $S_{\mathbf{x}}$ and S_{σ} of pure strategies (singletons in Algorithm 1) for the x-player and the σ-player, and then grows those sets in every iteration by applying the best-response oracles to the current NE of the game G restricted to pure strategies in $S_{\mathbf{x}}$ and S_{σ}. At each iteration, an NE (in mixed strategy) of the restricted game is computed via linear program $\mathcal{P}_{FO}$ (where Σ is replaced by S_{σ} in the definition of constraints 3 and $\mathcal{X}$ is replaced by $S_{\mathbf{x}}$ in the last line). Convergence is achieved when the best-response oracles generate pure strategies that are already in sets $S_{\mathbf{x}}$ and S_{σ}. In other words, convergence is obtained if for the current NE both players cannot improve their strategies by looking outside the restricted game.

Algorithm 1: Double Oracle Algorithm

Data: Singletons $S_{\mathbf{x}}=\{\mathbf{x}\}$ and $S_{\sigma}=\{\sigma\}$ including an arbitrary solution and an arbitrary permutation
Result: a (possibly mixed) NE
converge $\leftarrow$ False
while *converge is False* **do**
  Find Nash equilibrium $(P_{\mathcal{X}}, P_{\Sigma}) \in G = (S_{\mathbf{x}}, S_{\sigma})$
  Find $\mathbf{x} = BR_{\mathbf{x}}(P_{\Sigma})$ and $\sigma = BR_{\sigma}(P_{\mathcal{X}})$
  if $\mathbf{x} \in S_{\mathbf{x}}$ *and* $\sigma \in S_{\sigma}$ **then** converge $\leftarrow$ True **else** add $\mathbf{x}$ to $S_{\mathbf{x}}$ and σ to S_{σ}
return $(P_{\mathcal{X}}, P_{\mathcal{U}})$

The correctness of the double oracle algorithm for finite two-player zero-sum games has been established by McMahan et al. [18]; the intuition for this correctness is as follows. Once the algorithm converges, the current solution must be an equilibrium of the game, because each player's current strategy is a best response to the other player's current strategy. This stems from the fact that the best-response oracle, which searches over all possible strategies, cannot find anything better. Furthermore, the algorithm must converge, because at worst, it will generate all pure strategies.

Best response procedures. We now specify the procedures $BR_{\mathbf{x}}(\cdot)$ and $BR_\sigma(\cdot)$ used in our double oracle algorithm. Given a mixed strategy P_Σ of the σ-player, $BR_{\mathbf{x}}(P_\Sigma)$ is a pure solution which maximizes $\sum_{\sigma\in\Sigma} P_\Sigma(\sigma)\mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(\mathbf{x}))$, which can be rewritten as:

$$\sum_{\sigma\in\Sigma} P_\Sigma(\sigma)\sum_{i=1}^n w_{\sigma(i)}u_i(\mathbf{x}) = \sum_{i=1}^n(\sum_{\sigma\in\Sigma} P_\Sigma(\sigma)w_{\sigma(i)})u_i(\mathbf{x}).$$

Thus, computing $BR_{\mathbf{x}}(P_\Sigma)$ amounts to solving problem $\mathcal{P}$ according to a WA criterion with weights $\tilde{\sigma}$ defined by $\tilde{\sigma}_i = \sum_{\sigma\in\Sigma} P_\Sigma(\sigma)w_{\sigma(i)}$. Note that, as optimal solutions according to WA criteria are Pareto optimal, Pareto-dominated solutions will not be generated by the algorithm. In other words, there will be no irrelevant feasible solution added to set $S_{\mathbf{x}}$.

We now turn to the best response procedure for the σ-player. Given a mixed strategy $P_{\mathcal{X}}$ of the x-player, $BR_\sigma(P_{\mathcal{X}})$ is a permutation in Σ which minimizes $\mathtt{WA}_{\mathbf{w}_\sigma}(\mathbf{u}(P_{\mathcal{X}}))$. Similarly to the separation oracle in the cutting plane method for solving $\widetilde{\mathcal{P}}_{FO}$, best response $BR_\sigma(P_{\mathcal{X}})$ can be computed by sorting vector $\mathbf{u}(P_{\mathcal{X}})$. Interestingly, procedure $BR_\sigma(\cdot)$ is independent of the problem $\mathcal{P}$ considered.

There is no polynomial guarantee on the number of iterations of this double oracle algorithm. It will nevertheless reveal efficient in practice, as will be shown by the numerical tests presented in the next section.

4 Numerical Tests

We tested our approaches on the one-to-one assignment problem with a number n of agents varying from 50 to 300. Given an assignment $\mathbf{x}$, the utility $u_i(\mathbf{x})$ is given by the expression $\sum_{j=1}^n u_{ij}x_{ij}$ where $x_{ij} = 1$ if agent i receives object j (0 otherwise) and u_{ij} is a scalar value giving the utility for agent i to receive object j. For this problem, program $\widetilde{\mathcal{P}}_{FO}$ reads as follows (FOA for Fair Optimization Assignment):

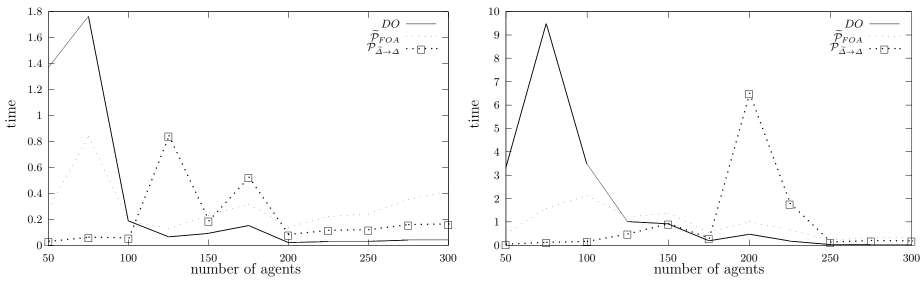
$$\widetilde{\mathcal{P}}_{FOA}\begin{cases} \max\limits_{v,x_{11},\ldots,x_{nn}} \quad v \\ v \le \sum\limits_{i=1}^n p_{\sigma(i)}\sum\limits_{j=0}^n x_{ij}u_{ij} & \forall\sigma\in\Sigma \\ \sum\limits_{j=1}^n x_{ij} = 1 & \forall i\in\{1,\ldots,n\} \\ \sum\limits_{i=1}^n x_{ij} = 1 & \forall j\in\{1,\ldots,n\} \\ x_{ij}\ge 0 & \forall i,j\in\{1,\ldots,n\} \end{cases}$$

Both the separation oracle for $\mathcal{P}_{\widetilde{\Delta}\to\Delta}$ and the procedure $BR_{\mathbf{x}}(\cdot)$ in the double oracle algorithm require to solve a standard assignment problem, which can be performed in polynomial time using the Hungarian method [7].

Performances in computation times. We compare the solution times of program $\widetilde{\mathcal{P}}_{FOA}$, program $\mathcal{P}_{\widetilde{\Delta}\to\Delta}$ and the double oracle algorithm (abbreviated by DO in the figures). For this purpose, we carried out numerical tests[2] on

[2] All methods were coded in C++ using Gurobi 5.6.3 as solver for the linear programs. Times are wall-clock times on a 2.4 GHz Intel Core i5 machine with 8GB of RAM.

randomly generated instances of increasing sizes. The weights w_i used are the ones given in Equation 1 and the utilities u_{ij} are uniformly drawn as positive integer values in $[1, U]$. In Figure 2(a), U is set to 20 while in Figure 2(b) it is set to 30. For all values of n and U considered, the average computation times are less than 10 seconds, which shows the practicality of the methods. On both figures, we observe that the computation time first increases, reaches a peak and then decreases. This phenomenon may be seen as a *phase transition*: if $n \gg U$ it is likely that there will exist a pure assignment x such that $u_i(x) = U$ for all agents i. In that case, the three methods $\widetilde{\mathcal{P}}_{FOA}$, $\mathcal{P}_{\widetilde{\Delta}\to\Delta}$ and DO will converge in very few iterations. In both figures, we observe that the sequence of the peaks is identical: DO first, $\widetilde{\mathcal{P}}_{FOA}$ second, and $\mathcal{P}_{\widetilde{\Delta}\to\Delta}$ third. Note that DO and $\mathcal{P}_{\widetilde{\Delta}\to\Delta}$ seem to be more affected by an increase of U than $\widetilde{\mathcal{P}}_{FOA}$. This can be explained by the fact that both algorithms rely on an oracle method (the Hungarian method) the complexity of which is $O(n^3)$, while the oracle used in $\widetilde{\mathcal{P}}_{FOA}$ is simply a sorting algorithm. Therefore DO and $\mathcal{P}_{\widetilde{\Delta}\to\Delta}$ are more sensitive to an increase of the number of calls to oracles. Lastly, the graphs clearly show that the choice of the algorithm to use should be made according to the ratio n/U. If this ratio is low (less than 5 for instance), one should prefer the combination of $\widetilde{\mathcal{P}}_{FOA}$ and $\mathcal{P}_{\widetilde{\Delta}\to\Delta}$, while if it is higher, DO will be more efficient.



(a) Costs are generated in $[1, 20]$. (b) Costs are generated in $[1, 30]$.

Fig. 2: Computation time (in seconds) as n increases. Results averaged on 20 instances.

5 Conclusion

We tackled the randomized version of fair multi-agent optimization problems with a generalized Gini index as optimization criterion. Thanks to a game-theoretic view of this problem, we proposed two solution methods based on dynamic calls to oracles. The first method we studied (cutting plane method) is polynomial time in p and n if the single objective optimization version is polynomial in p. The second method we studied (double oracle method) has the advantage to be operational for a broader class of problems. The numerical tests carried out show the practicality of both methods. For future works, it would be interesting to extend our results to fair multi-agent optimization problems with Choquet integrals. Indeed, OWA operators are special cases of Choquet integrals.

Acknowledgments This work is supported by the ANR project CoCoRICo-CoDec ANR-14-CE24-0007-01.

References

1. Abdulkadiroğlu, A., Sönmez, T.: Random serial dictatorship and the core from random endowments in house allocation problems. Econometr. 66(3), 689–701 (1998)
2. Aziz, H., Brandt, F., Stursberg, P.: On popular random assignments. In: Inter. Symp. on Algorithmic Game Theory. pp. 183–194. Springer (2013)
3. Blackorby, C., Donaldson, D.: Measures of relative equality and their meaning in terms of social welfare. Journal of Economic Theory 18(1), 59–80 (1978)
4. Bogomolnaia, A., Moulin, H.: A new solution to the random assignment problem. J. of Economic theory 100(2), 295–328 (2001)
5. Chong, K.M.: An induction theorem for rearrangements. Canadian J. of Math. 28, 154–160 (1976)
6. Chvatal, V.: Linear programming. Macmillan (1983)
7. Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: Combinatorial optimization, vol. 605. Springer (1998)
8. Endriss, U.: Reduction of economic inequality in combinatorial domains. In: Proceedings of AAMAS 2013. pp. 175–182 (2013)
9. Fodor, J., Marichal, J.L., Roubens, M.: Characterization of the ordered weighted averaging operators. IEEE Trans. on Fuzzy Systems 3(2), 236–240 (1995)
10. Goldsmith, J., Sloan, R.H.: The AI conference paper assignment problem. In: 2007 AAAI Workshop (2007)
11. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica 1(2), 169–197 (1981)
12. Katta, A.K., Sethuraman, J.: A solution to the random assignment problem on the full preference domain. Journal of Economic Theory 131(1), 231–250 (2006)
13. Kavitha, T., Mestre, J., Nasre, M.: Popular mixed matchings. Theoretical Computer Science 412(24), 2679–2690 (2011)
14. Lang, J., Rothe, J.: Fair division of indivisible goods. In: Economics and Computation, pp. 493–550. Springer (2016)
15. Lesca, J., Perny, P.: LP solvable models for multiagent fair allocation problems. In: ECAI. pp. 393–398 (2010)
16. Machina, M.J.: Dynamic consistency and non-expected utility models of choice under uncertainty. Journal of Economic Literature 27(4), 1622–1668 (1989)
17. Mastin, A., Jaillet, P., Chin, S.: Randomized minmax regret for combinatorial optimization under uncertainty. In: Int. Symp. on Algorithms and Computation. pp. 491–501. Springer (2015)
18. McMahan, H.B., Gordon, G.J., Blum, A.: Planning in the presence of cost functions controlled by an adversary. In: ICML. pp. 536–543 (2003)
19. Moulin, H.: Axioms of cooperative decision making. Cambridge Univ. Press (1991)
20. Ogryczak, W., Śliwiński, T.: On solving linear programs with the ordered weighted averaging objective. European J. of Operational Research 148(1), 80–91 (2003)
21. Skowron, P., Faliszewski, P., Lang, J.: Finding a collective set of items: From proportional multirepresentation to group recommendation. Artificial Intelligence 241, 191–216 (2016)
22. Skowron, P., Faliszewski, P., Slinko, A.: Achieving fully proportional representation: Approximability results. Artificial Intelligence 222, 67–103 (2015)

23. Weymark, J.A.: Generalized gini inequality indices. Mathematical Social Sciences 1(4), 409–430 (1981)
24. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Trans. on Sys., Man, and Cybernetics 18(1), 183–190 (1988)