Fair Proportional Representation Problems with Mixture Operators

Hugo Gilbert

▶ **To cite this version:**

Hugo Gilbert. Fair Proportional Representation Problems with Mixture Operators. 5th International Conference on Algorithmic Decision Theory (ADT 2017), Oct 2017, Luxembourg, Luxembourg. hal-01560630

HAL Id: hal-01560630
https://hal.science/hal-01560630

Submitted on 11 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fair Proportional Representation Problems with Mixture Operators

Hugo Gilbert(✉)

Sorbonne Universités, UPMC Univ Paris 06, CNRS,
LIP6 UMR 7606, 4 place Jussieu, 75005 Paris
hugo.gilbert@lip6.fr

Abstract. This paper deals with proportional representation problems in which a set of winning candidates must be selected according to the ballots of the voters. We investigate the use of a new class of optimization criteria to determine the set of winning candidates, namely mixture operators. In a nutshell, mixture operators are similar to weighted means where the numerical weights are replaced by weighting functions. In this paper: (1) we give the mathematical condition for which a mixture operator is fair and provide several instances of this operator satisfying this condition; (2) we show that when using a mixture operator as optimization criterion, one recovers the same complexity results as in the utilitarian case (i.e., maximizing the sum of agent's utilities) under a light condition; (3) we present solution methods to find an optimal set of winners w.r.t. a mixture operator under both Monroe and Chamberlin-Courant multi-winner voting rules and test their computational efficiency.

Keywords: Computational Social Choice · Inequality Measurement · Mixture Operators · Proportional Representation.

1 Introduction

This paper deals with multi-winner voting rules where one aims at electing a subset of candidates rather than a single one. In multi-winner election rules, a set of voters express preferences over a set of candidates. The objective is then to determine k winning candidates such that each voter is satisfied by the winning candidate that represents her. Multi-winner election rules are important for both political elections (i.e., electing committees of representatives) and multi-agent recommendation systems (e.g., choosing a set of dinning menus for a conference) [10,27]. A key property for multi-winner voting rules is Proportional Representation (PR), i.e., the proportional support enjoyed by the different candidates should be accurately reflected by the results of the elections.

Two multi-winner voting rules have been designed to account for PR, namely *Monroe*'s Voting Rule [19] (abbreviated by MVR) and *Chamberlin-Courant*'s Voting Rule [5] (abbreviated by CCVR). In these two frameworks, a feasible solution is characterized by a set of k winning candidates and an assignment from winning candidates to voters. Each voter is then represented by the elected

candidate assigned to her. While CCVR does not constraint the possible assignments from winning candidates to voters, MVR imposes that the k sets consisting of the voters represented by the same candidate should be equally sized. The choice of the solution is then based on the ballots, where each voter ranks the candidates from best to worst. Indeed, the solution chosen should maximize the utilities of the voters, where the utility (i.e., satisfaction level) of a voter depends on the rank she gave to the candidate assigned to her. In the utilitarian version of MVR and CCVR, the goal is to find a solution maximizing the sum of voter's utilities. However, maximizing the sum of utilities can yield an unfair solution as it compensates between the utilities of the different voters. Thus, several alternative optimization criteria have been investigated for multi-winner voting rules to address this problem. In CCVR, Betzler et al. proposed to maximize the utility of the least happy voter [4]. This is known as the egalitarian version of CCVR. However, maximizing the minimum utility value of the voters can be considered extreme as it does not take into account the satisfaction of all but one voter. To adress this issue, Elkind and Ismaili extended this approach to Ordered Weighted Averages (OWAs) of utilities, which provides a smooth interpolation between the egalitarian version and the utilitarian version of CCVR [11].

In this paper, we investigate the use of another aggregation operator, namely Mixture Operators (MOs), to find an efficient and fair solution with CCVR and MVR. In a nutshell, MOs are similar to weighted means where the numerical weights are replaced by weighting functions. The solution sought with the MO should be efficient in the sense that the vector of voter's satisfactions should be Pareto optimal (i.e., the utility of a voter cannot be improved without decreasing the utility of another voter) and fair in the sense that the vector of voter's satisfactions should be well-balanced (which will be formalized later). MOs have recently been investigated in multi-criteria decision making. Indeed, while those operators (which are instances of Bajraktarević means) are not new, they have received a renewed interest due to successful applications in data fusion [1, 24].

Regarding the complexity of PR problems, Procaccia et al. proved that winner determination under the utilitarian versions of MVR and CCVR are both NP-hard problems even if the utility values are based on approval ballots [23]. Similarly, for the egalitarian version of CCVR, Betzler et al. proved that winner determination is NP-hard [4]. More positive results where obtained by resorting to approximation algorithms or special structures of preferences. Approximation algorithms for PR problems were given by Lu and Boutilier [16] and Skowron et al. [28]. Betzler et al. showed that, for single-peaked preferences, winner determination under CCVR is a polynomial time problem by designing a dynamic programming solution method [4]. Their results where extended by Cornaz et al. to the case of clustered single-peaked preferences [7]. Lastly, Skowron et al. showed that, for single-crossing preferences, winner determination under CCVR is also a polynomial time problem [29]. In this paper, we investigate the complexity of solving PR problems under CCVR and MVR with an MO as optimization criterion and show that we obtain the same complexity results as in the utilitarian case under a light condition.

The paper is organized as follows. Section 2 presents PR problems under MVR and CCVR, introduces our notations and discusses the use of several criteria to solve these problems. In Section 3, we present MOs and their properties. In Section 4, we design several solution procedures to solve PR problems with an MO criterion. Lastly, our numerical tests are presented in Section 5.

2 Proportional Representation

Let $V = \{1, \ldots, n\}$ be a set of n voters and $C = \{1, \ldots, m\}$ be a set of m candidates. A solution of the PR problem is a set of k winning candidates $\{c_1, \ldots, c_k\} \subseteq C$, and k sets S_j ($j \in \{c_1, \ldots, c_k\}$), where S_j is the subset of voters represented by candidate j. We recall that, while in MVR the sets S_j should be of the same size, it is not the case in CCVR. Consequently, in CCVR, a voter is always represented by the candidate she likes most in the set of k winning candidates. We will denote by $\mathbf{e} = \{c_1, S_{c_1}; \ldots; c_k, S_{c_k}\}$ any feasible solution and by $\mathcal{E}$ the set of all feasible solutions. Each voter has preferences over candidates. These preferences are expressed by a *preference profile* P of size mn, where the i^{th} column of P is the preference order of voter i. From profile P, we derive nm utility values v_{ij} that represent the level of satisfaction of voter i if she is represented by candidate j. Given a feasible solution $\mathbf{e}$, the utility $v_i(\mathbf{e})$ of voter i is then given by v_{ij} if i belongs to S_j in $\mathbf{e}$. The PR problem aims at determining a solution $\mathbf{e} \in \mathcal{E}$ such that the utilities of the voters are maximized. We assume that all utility values v_i belong to some open interval[1] $\mathtt{D} \subset \mathbb{R}^+$ and we denote by $\mathbf{v}(\mathbf{e}) = (v_1(\mathbf{e}), \ldots, v_n(\mathbf{e})) \in \mathtt{D}^n$ the vector[2] giving the utilities of each voter for solution $\mathbf{e}$. A solution $\mathbf{e}_1$ will then be preferred to another solution $\mathbf{e}_2$ if, globally, $\mathbf{e}_1$ satisfies more the voters than $\mathbf{e}_2$. This is formalized by an aggregation criterion to maximize. More formally, given an operator $F : \mathtt{D}^n \to \mathbb{R}$, a solution $\mathbf{e}_1$ is preferred to a solution $\mathbf{e}_2$ if $F(\mathbf{v}(\mathbf{e}_1)) \geq F(\mathbf{v}(\mathbf{e}_2))$. The problem of finding an optimal solution is then written as follows:

$$\max_{\mathbf{e} \in \mathcal{E}} \quad F(\mathbf{v}(\mathbf{e})) \tag{1}$$

The choice of the operator F is a key but difficult point. A "good" operator should both enable to solve efficiently the maximization problem defined by Equation 1 and ensure that the optimal solution found satisfies some desirable theoretical properties, as for instance *Pareto optimality*.

Definition 1 *Vector* $\mathbf{y} \in \mathtt{D}^n$ Pareto-dominates *vector* $\mathbf{y}' \in \mathtt{D}^n$ *if:*

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

A solution $\mathbf{e}$ *is said to be* Pareto-optimal *iff there is no* $\mathbf{e}' \in \mathcal{E}$ *such that* $\mathbf{v}(\mathbf{e}')$ *Pareto-dominates* $\mathbf{v}(\mathbf{e})$.

Pareto optimality ensures that the solution is efficient in the sense that the utility of a voter cannot be increased without decreasing the utility of another voter.

[1] Considering an open interval simplifies the writing of Proposition 1 in Section 3.
[2] Note that we follow the convention to use bold letters to represent vectors.

We will say that an operator F is Pareto-compatible if any optimal solution of maximization problem 1 w.r.t. operator F is Pareto-optimal. To ensure Pareto-compatibility, operator F should be strictly increasing in each of its variables. In this case, we will say that the operator is *increasing*. A simple class of increasing operators is the class of Weighted Averages (WA) (with strictly positive weights):

Definition 2 *Let* $\mathbf{w} = (w_1, \ldots, w_n)$ *be a vector of weights. The* $\mathtt{WA}_{\mathbf{w}}(\cdot)$ *operator induced by* $\mathbf{w}$ *is defined by:*

$$\forall \mathbf{x} \in \mathtt{D}^n, \mathtt{WA}_{\mathbf{w}}(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i$$

Given a WA operator, a solution $\mathbf{e}$ *is then evaluated by* $\mathtt{WA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}))$.

Optimizing a WA criterion is attractive due to the simplicity of this aggregation operator. In fact, the average operator used in the utilitarian version of PR problems is a WA operator, with $w_1 = \ldots = w_n$ in order to treat each voter identically. However, such criterion may lead to an unfair solution as it compensates between the utility values of the different voters. For instance, for two voters and $\mathbf{w} = (1, 1)$, the utility vector $(3, 10)$ is preferred to utility vector $(6, 6)$ for the $\mathtt{WA}_{\mathbf{w}}$ criterion. This is not satisfying as fairness should be an important property for multi-winner voting problems. A natural condition that can be satisfied by an operator to favor fairness is the Pigou-Dalton transfer principle [20]:

Definition 3 Pigou-Dalton Transfer Principle. *Let* $\mathbf{x} \in \mathtt{D}^n$ *such that* $x_i > x_j$ *for some* i, j. *Then, for all* ϵ *such that* $0 < \epsilon < x_i - x_j$, $\mathbf{x} - \epsilon\mathbf{b}_i + \epsilon\mathbf{b}_j$ *should be strictly preferred to* $\mathbf{x}$ *where* $\mathbf{b}_i$ *and* $\mathbf{b}_j$ *are the vector whose ith (resp. jth) component equals 1, all others being null.*

The transfer principle states that a transfer from a "more satisfied" voter to a "less satisfied" voter should improve a solution. Indeed, such a transfer reduces inequality while keeping the arithmetic mean of the vector constant. We will say that an operator is *fair* if it satisfies the Pigou-Dalton transfer principle.

A well known class of fair operators whose optimization yields a Pareto-optimal solution is the class of *Ordered Weighted Average* (OWA) operators with strictly decreasing weights [32].

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:*

$$\forall \mathbf{x} \in \mathtt{D}^n, \mathtt{OWA}_{\mathbf{w}}(\mathbf{x}) = \sum_{i=1}^{n} w_i x_{\sigma(i)}$$

where σ *is a permutation of* $\{1, \ldots, n\}$ *such that* $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \ldots \leq x_{\sigma(n)}$. *Given an OWA operator, a solution* $\mathbf{e}$ *is then evaluated by* $\mathtt{OWA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}))$.

Note that both the average operator used in the utilitarian version of PR problems and the min operator used in the egalitarian version of PR problems are OWA operators obtained by setting $\mathbf{w} = (1, \ldots, 1)$ and $\mathbf{w} = (1, 0, \ldots, 0)$ respectively. PR problems with OWA operators as optimization criteria have been investigated by Elkind and Ismaili [11]. The authors investigated several classes of weights $\mathbf{w}$ defining families of OWA operators that allows for a compromise between the utilitarian and the egalitarian versions of PR problems under CCVR.

Interestingly, they showed that if the preferences of the voters present particular structures (e.g., single-crossing, single-peaked), then it becomes possible to design polynomial or pseudo-polynomial solution methods for some of these OWA families. However, the validity conditions of these methods and the algorithms themselves depend on the family of weights used.

In the next section, we present the class of operators that we will use as optimization criteria, namely *mixture operators*. Contrary to OWA operators, the way mixture operators weight the different components of a vector do not require any reordering. After recalling the main properties of this class of operators, we will give the condition under which they are fair. Interestingly, we will see that mixture operators have some descriptive advantages over OWA operators.

3 Mixture Operators

In this section, we present Mixture Operators (MOs) and their relations to several other operators.

Definition 5 *Let $w : \mathtt{D} \to (0, \infty]$ be a positive weighting function. The* mixture operator $M_w(\cdot)$ *induced by function w is defined as follows:*

$$\forall \mathbf{x} \in \mathtt{D}^n, M_w(\mathbf{x}) = \sum_{i=1}^{n} \frac{w(x_i)}{\sum_{j=1}^{n} w(x_j)} x_i$$

Given a mixture operator, a solution $\mathbf{e} \in \mathcal{E}$ is then evaluated by $M_w(\mathbf{v}(\mathbf{e}))$.

MOs are special instances of Losonczi means [15], Bajraktarević means and generalized mixture functions [25]. On the other hand, MOs extend several averaging operators as the Gini mean or the Lehmer mean. MOs resemble OWA and WA operators but their weights depend on the *values* that are at stake in the evaluated vector and not on the *ranks* of the components. Note that if function w is constant then the MO boils down to the average operator used in the utilitarian version of PR problems. Furthermore, the special case of the Lehmer mean is defined by $w(x) = x^{p-1}$ where p is a parameter. If p tends towards $-\infty$, then the Lehmer mean tends towards the min operator used in the egalitarian version of PR problems. The following example illustrates the way an MO distorts the weights of the voters.

Example 1 *Let $\mathtt{D} = (0, U)$ and consider the weight function w defined by $w(x) = 2U - x$. Then, given a solution $\mathbf{e}$ of the PR problem, the weight w_i associated to voter i is given by:*

$$w_i = \frac{w(v_i(\mathbf{e}))}{\sum_{j=1}^{n} w(v_j(\mathbf{e}))} = \frac{2U - v_i(\mathbf{e})}{\sum_{j=1}^{n} 2U - v_j(\mathbf{e})} = \frac{2U - v_i(\mathbf{e})}{n(2U - E)}$$

where $E = \sum_{j=1}^{n} v_j(\mathbf{e})/n$ is the arithmetic mean of vector $\mathbf{v}(\mathbf{e})$. Thus, $w_i \geq$ (resp $\leq$) $1/n$ if $v_i(\mathbf{e}) \leq$ (resp. $\geq$) E. Stated differently, if a voter's utility is less than the average utility of the voters, then she is given a greater weight in operator M_w. More generally, the more w is decreasing, the more operator M_w will focus on the least satisfied voters. This property enables us to find compromises between the utilitarian and the egalitarian variants of PR problems.

Moreover, by using a weighting function w depending on voters' utilities, MOs are able to account for preferences that cannot be represented by OWA or WA operators, as illustrated by the following example.

Example 2 *Let $n=2$ and consider four solutions $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}'_1$ and $\mathbf{e}'_2$ such that:*

$$\mathbf{v}(\mathbf{e}_1) = (4,8) \qquad \mathbf{v}(\mathbf{e}_2) = (2,12)$$
$$\mathbf{v}(\mathbf{e}'_1) = (8,8) \qquad \mathbf{v}(\mathbf{e}'_2) = (6,12).$$

Note that $\mathbf{v}(\mathbf{e}'_1)$ and $\mathbf{v}(\mathbf{e}'_2)$ are obtained from $\mathbf{v}(\mathbf{e}_1)$ and $\mathbf{v}(\mathbf{e}_2)$ by adding 4 to the utility value of the first voter. In solution $\mathbf{e}_2$, voter 1 is strongly unsatisfied, so one could prefer the more balanced solution $\mathbf{e}_1$ even if the average utility value of the voters is lower with $\mathbf{e}_1$ than with $\mathbf{e}_2$. Conversely, in both solutions $\mathbf{e}'_1$ and $\mathbf{e}'_2$, voters 1 and 2 are quite satisfied. Hence, one could prefer $\mathbf{e}'_2$ to $\mathbf{e}'_1$ as it yields the highest average utility value. Accounting for both of these preferences is not possible with a WA operator nor an OWA operator. Indeed, whatever the values of the weights $\mathbf{w} = (w_1, w_2)$:

$$\mathtt{WA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}'_1)) = \mathtt{WA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}_1)) + 4w_1 \quad \mathtt{WA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}'_2)) = \mathtt{WA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}_2)) + 4w_1$$
$$\mathtt{OWA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}'_1)) = \mathtt{OWA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}_1)) + 4w_1 \quad \mathtt{OWA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}'_2)) = \mathtt{OWA}_{\mathbf{w}}(\mathbf{v}(\mathbf{e}_2)) + 4w_1$$

Hence, if preferences are represented by a WA or an OWA, the preference holding between $\mathbf{e}_1$ and $\mathbf{e}_2$ should be the same as the one holding between $\mathbf{e}'_1$ and $\mathbf{e}'_2$. However, by considering $w(x) = 24 - x$, one obtains:

$$M_w(\mathbf{v}(\mathbf{e}_1)) \approx 5.78 \qquad M_w(\mathbf{v}(\mathbf{e}_2)) \approx 5.53$$
$$M_w(\mathbf{v}(\mathbf{e}'_1)) = 8 \qquad M_w(\mathbf{v}(\mathbf{e}'_2)) \approx 8.4$$

which is consistent with the desired preferences.

MOs have recently received some attention in multi-criteria decision making where their mathematical properties have been studied (e.g. monotonicity, orness, ...) [14,25].

Increasingness. Many works have been focused on the monotonicity of MOs [2,3]. Indeed, MOs are not increasing (and therefore not Pareto-compatible) in general. For this reason, sufficient conditions to ensure the monotonicity of this operator have been found. For instance, if $\mathtt{D} = (0, U)$ and if $w(x) \geq \frac{dw}{dx}(x)(U-x)$ for all $x \in (0, U)$ with w increasing and piecewise differentiable, then M_w is increasing [18]. A simpler condition to impose monotonicity, if $\mathtt{D} \subset \mathbb{R}^+$, is to have function $x \to w(x)$ decreasing and function $x \to w(x)x$ increasing.

Interestingly, MOs have also been studied in decision making under risk. Indeed, they are instances of both the Weighted Expected Utility (WEU) model [6] and the decomposable Skew-Symmetric Bilinear (SSB) functions investigated by Nakamura [21]. The properties of WEU functions and SSB functions w.r.t. risk-sensitivity and stochastic dominance have been thoroughly investigated [6,12,21] and these results can directly be used to entail results on MOs.

Fairness. We now give the condition under which MOs are fair (i.e., they satisfy the Pigou-Dalton transfer principle). This condition is a consequence of a result by Chew (Corollary 6 in [6]), who studied the consistency of WEU functions with stochastic dominance. Indeed, the Pigou-Dalton transfer principle in inequality

measurement coincides with consistency with second order stochastic dominance in decision making under risk.

Proposition 1 *([6]) Assume that functions $x \to w(x)$ and $x \to w(x)x$ as well as their first derivatives are continuous and bounded on $\mathtt{D}$, then operator $M_w(\cdot)$ is fair iff function $x \to w(x)(x-y)$ is strictly concave on $\mathtt{D}$ for all y in $\mathtt{D}$.*

Proof. For completeness, we give the sketch of the proof.
Sufficiency: Note that given vectors $\mathbf{x}, \mathbf{y} \in \mathtt{D}^n$:

$$\begin{aligned} M_w(\mathbf{x}) \geq M_w(\mathbf{y}) &\Leftrightarrow \sum_{i=1}^n \frac{w(x_i)}{\sum_{j=1}^n w(x_j)} x_i \geq \sum_{i=1}^n \frac{w(y_i)}{\sum_{j=1}^n w(y_j)} y_i \\ &\Leftrightarrow \sum_{i=1}^n \sum_{j=1}^n w(y_j)w(x_i)x_i \geq \sum_{i=1}^n \sum_{j=1}^n w(y_i)w(x_j)y_i \\ &\Leftrightarrow \sum_{i=1}^n \sum_{j=1}^n w(y_j)w(x_i)(x_i - y_j) \geq 0 \end{aligned}$$

In particular, for any vector $\mathbf{x} \in \mathtt{D}^n$,

$$\sum_{i=1}^n \sum_{j=1}^n w(x_j)w(x_i)(x_i - x_j) = 0.$$

Assume that function $x \to w(x)(x-y)$ is strictly concave on $\mathtt{D}$ for all y in $\mathtt{D}$, then function $\phi_{\mathbf{y}} : x \to \sum_{j=1}^n w(y_j)w(x)(x-y_j)$ is also strictly concave on $\mathtt{D}$ for all $\mathbf{y}$ in $\mathtt{D}^n$ as $w(y) > 0$ for all y in $\mathtt{D}$. Hence, by Lemma 2 in [8], if $\mathbf{x}^\epsilon$ is obtained from $\mathbf{x} \in \mathtt{D}^n$ by an ϵ-transfer (i.e., $\mathbf{x}^\epsilon = \mathbf{x} + \epsilon(\mathbf{b}_j - \mathbf{b}_i)$ with $0 < \epsilon < x_i - x_j$ and where $\mathbf{b}_i$ is the i^{th} canonical vector) then:

$$\sum_{i=1}^n \sum_{j=1}^n w(x_j)w(x_i^\epsilon)(x_i^\epsilon - x_j) > \sum_{i=1}^n \sum_{j=1}^n w(x_j)w(x_i)(x_i - x_j) = 0$$

Thus, $M_w(\mathbf{x}^\epsilon) > M_w(\mathbf{x})$ and the MO satisfies the Pigou-Dalton transfer principle.
Necessity: We recall that $\mathtt{D}$ is an open interval. By contradiction, assume there exists $s \in \mathtt{D}$ such that $x \to w(x)(x-s)$ is not strictly concave. Thus, there exists $\hat{x}$ and $\epsilon > 0$ such that $[\hat{x} - \epsilon, \hat{x} + \epsilon] \subset \mathtt{D}$ and $\phi_s : x \to w(s)w(x)(x-s)$ is convex on $[\hat{x} - \epsilon, \hat{x} + \epsilon]$. Assume w.l.o.g. that $s > \hat{x}$ and consider $t \in \mathtt{D}$ such that $t > s$. As $\mathbb{Q}$ is dense in $\mathbb{R}$ and as w is bounded and continuous on $\mathtt{D}$, t can be chosen such that there exists $k, l \in \mathbb{N}^*$ with $\phi_s(\hat{x}) + (k/l)\phi_s(t) = 0$. Set $n = 2(l+k)$ and consider vectors $\mathbf{x} = (\underbrace{\hat{x}, \hat{x}, \ldots, \hat{x}}_{2l}, \underbrace{t, \ldots, t}_{2k})$ and $\mathbf{s} = (\underbrace{s, \ldots, s}_{2(l+k)})$.
Then, by construction, $M_w(\mathbf{x}) = M_w(\mathbf{s})$. Consider $\mathbf{x}^\epsilon$ obtained by transferring ϵ from one of the $\hat{x}$ terms to another (increasing inequality). As in the sufficiency part, as ϕ_s is convex on $[\hat{x} - \epsilon, \hat{x} + \epsilon]$, we have $M_w(\mathbf{x}^\epsilon) \geq M_w(\mathbf{s}) = M_w(\mathbf{x})$ which violates the Pigou-Dalton transfer principle and concludes the proof. □

The conditions of Proposition 1 can easily be met. For instance, a sufficient condition, if $\mathtt{D} \subset \mathbb{R}^+$, is to have function $x \to w(x)x$ concave and function $x \to w(x)$ convex (with at least one property being strict). Under this sufficient condition, it is easy to see that the MO will be fair, as a Pigou-Dalton transfer will increase (resp. decrease) $\sum_{i=1}^n w(x_i)x_i$ (resp. $\sum_{i=1}^n w(x_i)$).

If $\mathtt{D} = (0, U)$, examples of MOs that are increasing and fair on $\mathtt{D}$ can be defined by using $w(x) = 1/(1+x)$ or $w(x) = (\alpha+1)U^\alpha - x^\alpha$ with $0 < \alpha \leq 1$. Indeed, in these cases, function $x \to w(x)$ is convex and decreasing on $\mathtt{D}$ and function $x \to w(x)x$ is strictly concave and increasing on $\mathtt{D}$.

In the next section, we investigate the complexity of winner determination in multi-winner voting rules with mixture operators.

4 Complexity and Solution Methods

Let u denote the function defined on $\mathtt{D}$ by $u(x) = w(x)x$. By abuse of notation, given a vector $\mathbf{x} \in \mathtt{D}^n$, we denote by $u(\mathbf{x})$ the sum $\sum_{i=1}^{n} u(x_i)$ and by $w(\mathbf{x})$ the sum $\sum_{i=1}^{n} w(x_i)$. By using these notations, the definition of an MO $M_w(\cdot)$ can be rewritten as follows:

$$\forall \mathbf{x} \in \mathtt{D}^n, M_w(\mathbf{x}) = \frac{\sum_{i=1}^{n} u(x_i)}{\sum_{j=1}^{n} w(x_j)} = \frac{u(\mathbf{x})}{w(\mathbf{x})} \tag{2}$$

Thus optimizing an MO entails the maximization of a ratio. Fractional programming is a subfield of operational research dedicated to this type of objective functions [9, 26, 30]. Several solution methods and techniques have been developed in this domain to optimize objective functions taking the form of a ratio of two linear objective functions. In this section, we will adapt and present two of these methods. Note that the two algorithms we present could also be used with the more general class of Losonczi means. The first one relies on a linearization trick that we will use to design a mixed integer linear program to solve PR problems w.r.t. an MO. The second one is a parametric approach that we will use to design polynomial time algorithms to solve PR problems with an MO criterion when a special structure of preferences makes it possible to solve the utilitarian version of the PR problem in polynomial time.

4.1 A Mixed Integer Linear Program

Note that if function w is constant, then one recovers the utilitarian version of PR problems which are NP-hard. Thus, it follows that PR problems under CCVR or MVR with an MO are also NP-hard. Yet, the NP-hardness of winner determination under CCVR and MVR has not prevented researchers from designing solution procedures for these problems. Indeed, Brams and Potthoff investigated the use of integer linear programs to solve PR problems [22]. We denote by $\mathcal{IP}$ the integer program they proposed. Program $\mathcal{IP}$ is given below on the left side of the page. It includes nm binary variables x_{ij} where x_{ij} takes value 1 if voter i is represented by candidate j and m binary variables z_j where z_j takes value 1 if candidate j represents at least one voter. Constraint (4) ensures that the election has k winning candidates. The n constraints in (5) make sure that each voter is only represented by one candidate. Lastly, constraints (6) and (7) specify a lower bound L and an upper bound U on the number of voters that can be represented by the same candidate. The pair (L, U) in MVR (resp. CCVR) is equal to $(\lfloor n/k \rfloor, \lceil n/k \rceil)$ (resp. $(0, n)$).

Program $\mathcal{IP}$ can be adapted to obtain a Mixed Integer Linear Program (MILP) $\mathcal{MIP}^{MO}$ (given below on the right side of the page) to solve a PR problem w.r.t. an MO. We now describe how it has been obtained. With the MO M_w, the objective function becomes:

$$\frac{\sum_{i,j \in V \cdot C} u(v_{ij}) x_{ij}}{\sum_{i,j \in V \cdot C} w(v_{ij}) x_{ij}} \tag{3}$$

$$\mathcal{IP}\begin{cases}\max \sum_{i,j\in V\cdot C} v_{ij}x_{ij} & \\ \sum_{j\in C} z_j = k & (4)\\ \sum_{j\in C} x_{ij} = 1, \quad \forall i \in V & (5)\\ \sum_{i\in V} x_{ij} \geq L z_j, \quad \forall j \in C & (6)\\ \sum_{i\in V} x_{ij} \leq U z_j, \quad \forall j \in C & (7)\\ z_j \in \{0,1\}, \quad \forall j \in C & \\ x_{ij} \in \{0,1\}, \quad \forall i,j \in V\cdot C & \end{cases}$$

$$\mathcal{MIP}^{MO}\begin{cases}\max \lambda & \\ \sum_{j\in C} z_j = k & \\ \sum_{j\in C} x_{ij} = 1, \quad \forall i \in V & \\ \sum_{i\in V} x_{ij} \geq L z_j, \quad \forall j \in C & \\ \sum_{i\in V} x_{ij} \leq U z_j, \quad \forall j \in C & \\ \sum_{i,j\in V\cdot C} (w(v_{ij})y_{ij} - u(v_{ij})x_{ij}) = 0 & \\ \sum_{j\in C} y_{ij} = \lambda, \quad \forall i \in V & \\ y_{ij} \leq x_{ij}\lambda^u, \quad \forall i,j \in V\cdot C & \\ z_j \in \{0,1\}, \quad \forall j \in C & \\ x_{ij} \in \{0,1\}, \quad \forall i,j \in V\cdot C & \\ y_{ij} \in \mathbb{R}^+, \quad \forall i,j \in V\cdot C & \\ \lambda \in \mathbb{R} & \end{cases}$$

To linearize this objective function, we use the following general method proposed by Williams [31]. Introduce a continuous variable λ into the problem to represent the expression given in Equation 3. The objective is then to maximize this variable. By definition of λ, the following condition should hold:

$$\sum_{i,j\in V\cdot C} w(v_{ij})\lambda x_{ij} - \sum_{i,j\in V\cdot C} u(v_{ij})x_{ij} = 0$$

However, this equation is not linear in the variables of the problem because of the quadratic terms λx_{ij}. Thus, to enforce this equation in program $\mathcal{MIP}^{MO}$, we introduce nm continuous variables y_{ij} taking values in $\mathbb{R}^+$ to replace expressions λx_{ij}. We then impose that $y_{ij} = \lambda x_{ij}$ with the following constraints:

$$y_{ij} \leq x_{ij}\lambda^u, \qquad \forall i,j \in V \cdot C \tag{8}$$

$$\sum_{j\in C} y_{ij} = \lambda, \qquad \forall i \in V \tag{9}$$

where λ^u denotes an upper bound on λ. Such an upper bound can easily be obtained by computing $(\max u(v_{ij}))/(\min w(v_{ij}))$. While Eq. 8 ensures that $y_{ij} = 0$ if $x_{ij} = 0$, Eq. 9 ensures that $y_{ij} = \lambda$ if $x_{ij} = 1$. Indeed, in Eq. 9, only one of the y_{ij} is non null due to constraints 5 in program $\mathcal{IP}$ and Eq. 9 imposes that this variable equals λ. The final program $\mathcal{MIP}^{MO}$ involves $nm + 1$ additional continuous variables and $nm + n + 1$ additional constraints.

4.2 A Parametric Approach

We now assume that the preferences of the voters have a particular structure which makes it possible to solve the utilitarian version of the PR problem with

a polynomial time algorithm denoted by $\mathcal{A}$. This is for instance the case for single-peaked or single-crossing preferences with CCVR. Using algorithm $\mathcal{A}$, we provide two polynomial time methods to solve the PR problem w.r.t. an MO. The first one follows from a method proposed by Megiddo [17] and the second one is a cutting plane method. The only light condition required by these two methods on the MO is that M_w should be an increasing MO with a decreasing function w and an increasing function u. Indeed, both of these methods require to solve utilitarian versions of the PR problem with utilities defined by $\tilde{v}_{ij}^{\lambda} = u(v_{ij}) - \lambda w(v_{ij})$ ($\lambda \in \mathbb{R}^+$). The previous condition on the monotony of w and u ensures that utilities $\tilde{v}_{ij}^{\lambda}$ are consistent with the preferences of the voters (i.e., if voter i prefers candidate j_1 to candidate j_2 then $\tilde{v}_{ij_1}^{\lambda} \geq \tilde{v}_{ij_2}^{\lambda}$).

Megiddo's method. The first method we present follows from a general method designed by Megiddo [17]. This method is based on the following observation (recasted in our PR setting):

Observation 1 *Let $\lambda \in \mathbb{R}$ and consider utility values $\tilde{v}_{ij}^{\lambda} = u(v_{ij}) - \lambda w(v_{ij})$. Let $\mathbf{e}^{\lambda}$ and $v^{\lambda} = u(\mathbf{v}(\mathbf{e}^{\lambda})) - \lambda w(\mathbf{v}(\mathbf{e}^{\lambda}))$ be the optimal solution and the optimal value for the utilitarian version of the PR problem with utility values $\tilde{v}_{ij}^{\lambda}$. Then, the sign of v^{λ} is determined by the position of λ w.r.t. λ^*, where λ^* is defined as the optimal value w.r.t. M_w (i.e. $\lambda^* = \max_{\mathbf{e}\in\mathcal{E}} M_w(\mathbf{v}(\mathbf{e}))$).*

1. If $\lambda{=}\lambda^*$, then $v^{\lambda}{=}0$ and $\mathbf{e}^{\lambda}$ is an optimal solution according to MO M_w.

2. If $\lambda{>}\lambda^*$, then v^{λ} will be strictly negative. Indeed, there exists no feasible solution $\mathbf{e} \in \mathcal{E}$ such that $u(\mathbf{v}(\mathbf{e}))/w(\mathbf{v}(\mathbf{e})) > \lambda$.

3. On the contrary, if $\lambda{<}\lambda^*$, then v^{λ} will be strictly positive.

Thus, the problem of solving a PR problem w.r.t. an MO reduces to the one of solving the utilitarian version of the PR problem with utility values $\tilde{v}_{ij}^{\lambda^*} = u(v_{ij}) - \lambda^* w(v_{ij})$ (case 1 of Observation 1). However, while values $u(v_{ij})$ and $w(v_{ij})$ are known, the value of λ^* is not. Thus, the values $\tilde{v}_{ij}^{\lambda^*}$ are incompletely specified. Nevertheless, if the utilitarian version of the PR problem can be solved by an algorithm $\mathcal{A}$ relying on a polynomial number of additions/subtractions and comparisons (which is the case for PR problems under CCVR with single-peaked or single-crossing preferences), then this problem can be solved via Megiddo's method. In short, Megiddo's method applied to the PR problem mimics algorithm $\mathcal{A}$ to solve the utilitarian version of the PR problem with utility values $\tilde{v}_{ij}^{\lambda^*}$. However, the method redefines the addition and the comparison operations to handle the fact that λ^* is unknown.

Management of the imprecisely known value of λ^*. Instead of having precise values for $\tilde{v}_{ij}^{\lambda^*}$, the algorithm works with pairs of values $(u(v_{ij}), w(v_{ij}))$ for all i in V and j in C and maintains a lower bound λ^l and an upper bound λ^u over λ^* (originally 0 and ∞). Thus $\tilde{v}_{ij}^{\lambda^*}$ is only known to be in the interval $[u(v_{ij}) - \lambda^u w(v_{ij}), u(v_{ij}) - \lambda^l w(v_{ij})]$.

Redefinition of the addition operation. The additions and subtractions required by algorithm $\mathcal{A}$ are simply replaced by componentwise additions and subtractions of pairs. Stated differently, the sum of $(u(v_{ij}), w(v_{ij}))$ and $(u(v_{kl}), w(v_{kl}))$ is $(u(v_{ij}) + u(v_{kl}), w(v_{ij}) + w(v_{kl}))$. Indeed, whatever the value of λ^*:
$u(v_{ij}) - \lambda^* w(v_{ij}) + u(v_{kl}) - \lambda^* w(v_{kl}) = u(v_{ij}) + u(v_{kl}) - \lambda^*(w(v_{ij}) + w(v_{kl}))$.

Redefinition of the comparison operation. To compare two pairs $(u(v_{ij}), w(v_{ij}))$ and $(u(v_{kl}), w(v_{kl}))$, Megiddo's method uses the following routine.

A fist step consists in checking if $u(v_{ij}) - \lambda w(v_{ij})$ is less (resp. greater) than $u(v_{kl}) - \lambda w(v_{kl})$, regardless of the value of $\lambda \in [\lambda^l, \lambda^u]$. Indeed, in that case (illustrated on the left side of Figure 1 below), the algorithm can conclude that $u(v_{ij}) - \lambda^* w(v_{ij}) \leq$ (resp. $\geq$) $u(v_{kl}) - \lambda^* w(v_{kl})$. Note that $u(v_{ij}) - \lambda w(v_{ij})$ and $u(v_{kl}) - \lambda w(v_{kl})$ are linear functions of λ. Therefore, the method needs only to check the inequality for values λ^l and λ^u.

If this first step does not conclude which pair is the minimum (case illustrated on the right side of Figure 1), then the algorithm considers the value $\hat{\lambda}$ such that $u(v_{ij}) - \hat{\lambda} w(v_{ij}) = u(v_{kl}) - \hat{\lambda} w(v_{kl})$. Then, if we assume w.l.o.g.[3] that $u(v_{ij}) - \lambda^l w(v_{ij}) < u(v_{kl}) - \lambda^l w(v_{kl})$, we have:

$$\forall \lambda \in [\lambda^l, \hat{\lambda}], u(v_{ij}) - \lambda w(v_{ij}) \leq u(v_{kl}) - \lambda w(v_{kl}) \tag{10}$$

$$\forall \lambda \in [\hat{\lambda}, \lambda^u], u(v_{ij}) - \lambda w(v_{ij}) \geq u(v_{kl}) - \lambda w(v_{kl}). \tag{11}$$

Now, for the algorithm to conclude, it needs only to check if $\lambda^* \in [\lambda^l, \hat{\lambda}]$ (in which case, λ^u is set to $\hat{\lambda}$) or $\lambda^* \in [\hat{\lambda}, \lambda^u]$ (in which case, λ^l is set to $\hat{\lambda}$). This is done by testing the sign of the optimal value of the utilitarian version of the PR problem with utility values $\tilde{v}_{ij}^{\hat{\lambda}}$ (see Obs. 1), which is computed by using algorithm $\mathcal{A}$. Note that this operation updates either λ^l or λ^u, which refines the value of λ^* and enables us to perform more comparisons without rerunning algorithm $\mathcal{A}$.

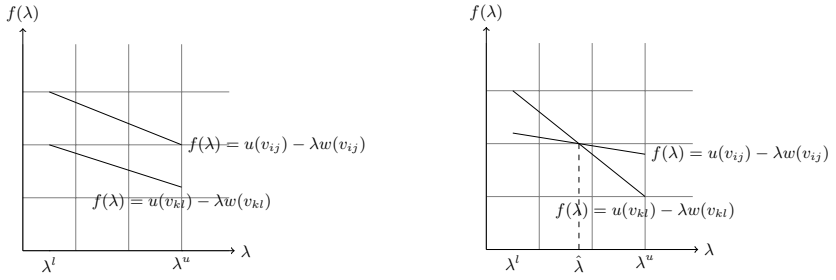

Fig. 1. Illustration of the two possible cases that can occur in the comparison routine.

Polynomial time complexity of the method. As Megiddo's method mimics algorithm $\mathcal{A}$, it relies on a polynomial number of (redefined) additions and compar-

[3] if, $u(v_{ij}) - \lambda^l w(v_{ij}) > u(v_{kl}) - \lambda^l w(v_{kl})$, just reverse inequalities 10 and 11.

isons. As algorithm $\mathcal{A}$ (which is used in the comparison operator) is of polynomial complexity, these two operations can be performed in polynomial time. This proves the polynomial complexity of the method.

Cutting plane method. As shown by Equation 2, for a solution $\mathbf{e}^*$ maximizing $M_w(\mathbf{v}(\mathbf{e}))$, we have $\frac{u(\mathbf{v}(\mathbf{e}^*))}{w(\mathbf{v}(\mathbf{e}^*))} \geq \frac{u(\mathbf{v}(\mathbf{e}))}{w(\mathbf{v}(\mathbf{e}))}$ for all $\mathbf{e}$ in $\mathcal{E}$. Replacing $u(\mathbf{v}(\mathbf{e}^*))/w(\mathbf{v}(\mathbf{e}^*))$ by λ^*, these inequalities rewrite as follows:

$$u(\mathbf{v}(\mathbf{e})) - \lambda^* w(\mathbf{v}(\mathbf{e})) \leq 0, \qquad \forall \mathbf{e} \in \mathcal{E} \tag{12}$$

and the constraint is tight for $\mathbf{e} = \mathbf{e}^*$. Therefore, λ^* is the minimal value such that all inequalities in 12 are satisfied. This analysis yields the following Linear Program (LP) $\mathcal{LP}^{MO}$:

$$\mathcal{LP}^{MO} \begin{cases} \min\limits_{\lambda} \lambda \\ u(\mathbf{v}(\mathbf{e})) - \lambda w(\mathbf{v}(\mathbf{e})) \leq 0 \qquad \forall \mathbf{e} \in \mathcal{E} \\ \lambda \in \mathbb{R} \end{cases} \tag{13}$$

Even if the number of constraints in 13 may be exponential in m (as there is one constraint per feasible solution), these constraints can be handled efficiently by resorting to a *cutting plane algorithm*. A cutting plane algorithm makes it possible to solve an LP involving a set $\mathcal{S}$ of constraints, the size of which is exponential, provided there exists a *separation oracle*. A separation oracle dynamically generates a violated constraint in $\mathcal{S}$ given the current optimal solution according to the previously generated constraints, or states that there is no violated constraint (in which case the current solution is optimal). In program $\mathcal{LP}^{MO}$, a separation oracle to determine a most violated constraint amounts to solve the utilitarian version of the PR problem with utilities defined by $\tilde{v}_{ij}^{\lambda} = u(v_{ij}) - \lambda w(v_{ij})$. This problem can be solved by algorithm $\mathcal{A}$. As the complexity of $\mathcal{A}$ is polynomial, the complexity of solving $\mathcal{LP}^{MO}$ is polynomial by the polynomial time equivalence of optimization and separation by resorting to the ellipsoid method [13].

In practice, one can resort to Dinkelbach's method [9] for solving program $\mathcal{LP}^{MO}$. Indeed, while Dinkelbach's method has not a polynomial time guarantee, it reveals more efficient in practice. In the setting of PR problems, this method can be described as follows. Let $\mathbf{e}^{\lambda}$ denote an optimal solution in $\mathcal{E}$ for the utilitarian variant of the PR problem with utilities $\tilde{v}_{ij}^{\lambda}$ (which can be obtained via algorithm $\mathcal{A}$). Program $\mathcal{LP}^{MO}$ can be solved by computing a sequence of solutions in $\mathcal{E}$ through the following recursive equation:

$$\mathbf{e}_{t+1} = \mathbf{e}^{\lambda_t}$$

where $\lambda_t = u(\mathbf{v}(\mathbf{e}_t))/w(\mathbf{v}(\mathbf{e}_t)) = M_w(\mathbf{v}(\mathbf{e}_t))$. The key point of this approach is that the solutions generated in this way are of increasing values w.r.t the MO operator. Indeed, a direct corollary from Observation 1 is that while $M_w(\mathbf{v}(\mathbf{e}_t)) < \lambda^*$, $M_w(\mathbf{v}(\mathbf{e}_{t+1})) > M_w(\mathbf{v}(\mathbf{e}_t))$. Note that, by definition of λ^*, we cannot have $M_w(\mathbf{v}(\mathbf{e}_t)) > \lambda^*$. Therefore, the sequence $(M_w(\mathbf{v}(\mathbf{e}_t)))_{t\in\mathbb{N}}$ is strictly increasing until reaching λ^*. Value λ^* is always reached after a finite number of iterations as there is a finite number of values in $\{M_w(\mathbf{v}(\mathbf{e})) : \mathbf{e} \in \mathcal{E}\}$. After a finite number

of iterations, we will thus have $M_w(\mathbf{v}(\mathbf{e}_t)) = M_w(\mathbf{v}(\mathbf{e}_{t+1}))$ which means that an optimal solution w.r.t. the MO has been found.

We now turn to our numerical tests in which the efficiency of the different solution methods are compared.

5 Numerical Tests

In this section, we compare the execution times of the proposed solution methods in two different experiments[4]. In both experiments, values v_{ij} are set to $m - \mathtt{rk_i(j)}$ (where $\mathtt{rk_i(j)}$ denotes the rank of candidate j in the preference order of voter i) and the weighting function w of the MO is defined by $w(x) = 2m - x$, so that the transfer principle holds. Lastly, for an election with n voters, values m and k were set to $n/5$ and $n/20$.

In the first experiment, we restrict ourselves to randomly generated single-peaked preferences in the CCVR framework, and we compare the execution times of the two methods presented in section 4.2, denoted by CP (for Cutting Plane) and MEG (for Megiddo). The method CP is solved by using the Dinkelbach method. In methods CP and MEG, the algorithm $\mathcal{A}$ used (see section 4.2) is the dynamic programming algorithm proposed by Betzler et al. [4]. The average computation times (in seconds) over 50 instances, as well as the average number of calls to algorithm $\mathcal{A}$, are given in Table 1 for instances with a number of voters ranging from 100 to 1000.

We observe that both methods CP and MEG are very fast and require very few calls to algorithm $\mathcal{A}$. On these instances, method CP seems to perform best and requires less calls to algorithm $\mathcal{A}$.

Table 1. Evolution of the average computation time in seconds and the average number of calls to algorithm $\mathcal{A}$ (nbc in the table) for methods CP and MEG as n increases.

n		100	200	300	400	500	600	700	800	900	1000
CP	time	<0.001	0.005	0.017	0.037	0.074	0.132	0.276	0.416	0.618	1.064
	nbc	2.62	2.86	2.94	3.00	2.98	2.98	3.00	3.00	3.00	3.00
MEG	time	0.001	0.017	0.064	0.171	0.370	0.676	1.461	2.360	3.872	6.981
	nbc	3.42	7.18	8.90	11.32	11.94	12.40	13.34	14.32	16.20	15.04

In a second experiment, we compare the execution times of the two instantiations of $\mathcal{MIP}^{MO}$ in the CCVR and MVR frameworks. We denote these programs by $\mathcal{M}^{MO}_{CCVR}$ and $\mathcal{M}^{MO}_{MVR}$. For each instance, the preferences of the voters were generated uniformly at random. The average computation times (in seconds) over 50 instances are given in Table 2 for instances with a number of voters ranging from 50 to 100.

Obviously, we observe that solving those two programs are computationally demanding. For instance, solving programs $\mathcal{M}^{MO}_{CCVR}$ and $\mathcal{M}^{MO}_{MVR}$ takes more than 25 seconds for elections with 100 voters.

[4] All methods were implemented in C++ using Gurobi version 5.6.3 to solve the LPs. Times are wall-clock times on a 2.4 GHz Intel Core i5 machine with 8GB of RAM.

Table 2. Evolution of the average computation time in seconds to solve programs $\mathcal{M}^{MO}_{CCVR}$ and $\mathcal{M}^{MO}_{MVR}$ as n increases.

n	50	60	70	80	90	100
$\mathcal{M}^{MO}_{CCVR}$	0.79	1.19	2.22	3.35	13.45	28.52
$\mathcal{M}^{MO}_{MVR}$	1.01	1.60	3.30	5.34	14.78	25.66

6 Conclusion

We studied PR problems with MOs. We presented these operators and the conditions under which they make it possible to find a fair solution for a PR problem. We designed a MILP to solve PR problems w.r.t. an MO and presented two other solution methods that are of polynomial complexity if the preferences of the voters abid to a particular structure enabling to solve the utilitarian version of the PR problem in polynomial time.

As future work, it would be worth investigating the extent to which the methods presented here can be adapted to other domains requiring fairness. For instance, it seems that they could also be used for the assignment problem. However, using MO for solving allocation problems where agents can receive several goods seems more challenging as the weighting function would be applied to a sum of utilities, which triggers new technical difficulties. Moreover, the results presented here can be extended to Losonczi means and we plan to investigate if those more general operators can yield new desirable properties for PR problems.

Acknowledgements. This work is supported by the ANR project CoCoRICoCoDec ANR-14-CE24-0007-01.

References

1. Angelov, P., Yager, R.: Density-based averaging–a new operator for data fusion. Information Sciences 222, 163–174 (2013)
2. Beliakov, G., Špirková, J.: Weak monotonicity of lehmer and gini means. Fuzzy sets and systems 299, 26–40 (2016)
3. Beliakov, G., Wilkin, T.: On some properties of weighted averaging with variable weights. Information Sciences 281, 1–7 (2014)
4. Betzler, N., Slinko, A., Uhlmann, J.: On the computation of fully proportional representation. Journal of Artificial Intelligence Research 47, 475–519 (2013)
5. Chamberlin, J.R., Courant, P.N.: Representative deliberations and representative decisions: Proportional representation and the borda rule. American Political Science Review 77(03), 718–733 (1983)
6. Chew, S.: A generalization of the quasilinear mean with applications to the measurement of income inequality and decision theory resolving the allais paradox. Econometrica: Journal of the Econometric Society pp. 1065–1092 (1983)
7. Cornaz, D., Galand, L., Spanjaard, O.: Bounded single-peaked width and proportional representation. In: Proceedings of ECAI 2012. pp. 270–275. IOS Press
8. Dasgupta, P., Sen, A., Starrett, D.: Notes on the measurement of inequality. Journal of economic theory 6(2), 180–187 (1973)
9. Dinkelbach, W.: On nonlinear fractional programming. Management science 13(7), 492–498 (1967)

10. Elkind, E., Faliszewski, P., Skowron, P., Slinko, A.: Properties of multiwinner voting rules. In: Proceedings AAMAS 2014
11. Elkind, E., Ismaili, A.: OWA-based extensions of the chamberlin–courant rule. In: Proceedings of ADT 2015. pp. 486–502. Springer
12. Fishburn, P.C.: Dominance in SSB utility theory. Journal of Economic Theory 34(1), 130–148 (1984)
13. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica 1(2), 169–197 (1981)
14. Liu, X.: The orness measures for two compound quasi-arithmetic mean aggregation operators. International Journal of Approximate Reasoning 51(3), 305–334 (2010)
15. Losonczi, L.: General inequalities for nonsymmetric means. aequationes mathematicae 9(2-3), 221–235 (1973)
16. Lu, T., Boutilier, C.: Budgeted social choice: From consensus to personalized decision making. In: IJCAI. vol. 11, pp. 280–286 (2011)
17. Megiddo, N.: Combinatorial optimization with rational objective functions. Mathematics of Operations Research 4(4), 414–424 (1979)
18. Mesiar, R., Špirková, J.: Weighted means and weighting functions. Kybernetika 42(2), 151–160 (2006)
19. Monroe, B.L.: Fully proportional representation. American Political Science Review 89(04), 925–940 (1995)
20. Moulin, H.: Axioms of cooperative decision making. No. 15, Cambridge University Press (1991)
21. Nakamura, Y.: Risk attitudes for nonlinear measurable utility. Annals of Operations Research 19(1), 311–333 (1989)
22. Potthoff, R.F., Brams, S.J.: Proportional representation: Broadening the options. Journal of Theoretical Politics 10(2), 147–178 (1998)
23. Procaccia, A.D., Rosenschein, J.S., Zohar, A.: On the complexity of achieving proportional representation. Social Choice and Welfare 30(3), 353–362 (2008)
24. Ribeiro, R.A., Falcão, A., Mora, A., Fonseca, J.M.: Fif: A fuzzy information fusion algorithm based on multi-criteria decision making. Knowledge-Based Systems 58, 23–32 (2014)
25. Ribeiro, R.A., Pereira, R.A.M.: Generalized mixture operators using weighting functions: A comparative study with WA and OWA. European Journal of Operational Research 145(2), 329–342 (2003)
26. Schaible, S., Ibaraki, T.: Fractional programming. European Journal of Operational Research 12(4), 325–338 (1983)
27. Skowron, P., Faliszewski, P., Lang, J.: Finding a collective set of items: From proportional multirepresentation to group recommendation. Artificial Intelligence 241, 191–216 (2016)
28. Skowron, P., Faliszewski, P., Slinko, A.: Achieving fully proportional representation: Approximability results. Artificial Intelligence 222, 67–103 (2015)
29. Skowron, P., Yu, L., Faliszewski, P., Elkind, E.: The complexity of fully proportional representation for single-crossing electorates. In: International Symposium on Algorithmic Game Theory. pp. 1–12. Springer (2013)
30. Stancu-Minasian, I.: Fractional programming: theory, methods and applications, vol. 409. Springer Science & Business Media (2012)
31. Williams, H.P.: Experiments in the formulation of integer programming problems. In: Approaches to Integer Programming, pp. 180–197. Springer (1974)
32. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Transactions on systems, Man, and Cybernetics (1988)