Noname manuscript No.
(will be inserted by the editor)

The Fair OWA One-to-one Assignment Problem: NP-hardness and Polynomial Time Special Cases.

Julien Lesca · Michel Minoux · Patrice Perny

Received: date / Accepted: date

Abstract We consider a one-to-one assignment problem consisting of matching n objects with n agents. Any matching leads to a utility vector whose n components measure the satisfaction of the various agents. We want to find an assignment maximizing a global utility defined as an ordered weighted average (OWA) of the n individual utilities. OWA weights are assumed to be non-increasing with ranks of satisfaction so as to include an idea of fairness in the definition of social utility. We first prove that the problem is NP-hard; then we propose a polynomial algorithm under some restrictions on the set of admissible weight vectors, proving that the problem belongs to XP.

Keywords Assignment problem · Fairness · Ordered Weighted Average · Complexity

1 Introduction

Fair multiagent optimization problems appear in various contexts such as resource allocation and sharing of indivisible goods [6, 27, 3, 23, 12, 15, 19], stable marriage problems [1], network optimization [28], and facility location [20, 2]. They motivate various studies at the interface of mathematical economics, decision theory

Julien Lesca
Université Paris-Dauphine, PSL Research University,
CNRS, LAMSADE,
F-75016 Paris, France.
E-mail: julien.lesca@dauphine.fr

Michel Minoux
Sorbonne Université, CNRS, Laboratoire d'informatique de Paris 6, LIP6,
F-75005 Paris, France.
E-mail: michel.minoux@lip6.fr

Patrice Perny
Sorbonne Université, CNRS, Laboratoire d'informatique de Paris 6, LIP6,
F-75005 Paris, France.
E-mail: patrice.perny@lip6.fr

and computer science. Some contributions study the axiomatic foundations of fairness and propose inequality measures, fair social evaluation functions, and more generally decision models including an idea of fairness [24,26,31,32,35]. Although the descriptive power of these models is now widely acknowledged, their use in preference-based combinatorial optimization remains a challenging question. On the one hand we need efficient algorithms to determine fair optimal solutions; on the other hand, the computational complexity of fair optimization remains an open question for some well known combinatorial problems. In particular, this is the case for the one-to-one assignment problem and this is what motivated the study proposed in this paper.

We consider a one-to-one assignment problem that consists in distributing a set of n objects $O = \{o_1, \dots, o_n\}$ to a set of n agents $A = \{a_1, \dots, a_n\}$ (one object per agent and one agent per object) while maximizing the overall satisfaction of the agents. It is assumed that the satisfaction of each individual agent only depends on the item he/she receives and is independent of the items assigned to the other agents (no externality). We assume here that individual preferences are represented by utility functions $u_i : O \to \mathbb{Z}$ where $u_i(o_j)$ is the utility of o_j for a_i. An instance of the multiagent assignment problem can thus be characterized by a square matrix U of size n where the term U_{ij} in line i and column j represents $u_i(o_j)$. For example, let us consider the following matrix where utility values are in the range $[0, 20]$:

$$U = \begin{pmatrix} 12 & 20 & 6 & 5 & 8 \\ 5 & 12 & 6 & 8 & 5 \\ 8 & 5 & 11 & 5 & 6 \\ 6 & 8 & 6 & 11 & 5 \\ 5 & 6 & 8 & 7 & 7 \end{pmatrix} \tag{1}$$

Formally, an assignment can be represented by a one-to-one mapping $\pi : A \to O$ where $\pi(a_i) = o_j$ when o_j is assigned to a_i. The assignment $\pi(a_1) = o_2, \pi(a_2) = o_1, \pi(a_3) = o_3$, $\pi(a_4) = o_4$ and $\pi(a_5) = o_5$ yields the utility vector $u(\pi) = (20, 5, 11, 11, 7)$. The assignment π maximizes the sum of the satisfactions of the agents (and therefore, the average satisfaction) with a sum equal to 54. It can easily be obtained with the Hungarian algorithm [21]. However we observe that this solution is not very fair because the satisfaction vector is ill-balanced. Hence other assignments leading to better balanced utility vectors $u(\pi') = (u_1(\pi'), \dots, u_n(\pi'))$ are worth considering.

As far as fairness is concerned, a more attractive utility profile is obtained for assignment π' defined by $\pi'(a_i) = o_i$ for $i = 1, \dots, 5$. Although π' is suboptimal in terms of average satisfaction (with a total score of 53), it could be checked that it is Pareto-optimal (no other assignment can improve one component without downgrading another one); moreover, the satisfaction vector $u(\pi') = (12, 12, 11, 11, 7)$ associated with π' is much better balanced than $u(\pi)$. When passing from π to π', we improve the value of the worst component, and obtain a more balanced profile with 4 agents above 10, while marginally decreasing the average. We may even wish to maximize the satisfaction of the least satisfied agent (egalitarianism). Under this criterion, the optimal solution would be $\pi''(a_1) = o_5, \pi''(a_2) = o_2, \pi''(a_3) = o_1$, $\pi''(a_4) = o_4$ and $\pi''(a_5) = o_3$ with $u(\pi'') = (8, 12, 8, 11, 8)$; every agent has utility of at least 8 but the average satisfaction is worse than for π or π'.

In our example, solution π' seems better than π and π''. It achieves a good tradeoff between utilitarianism (measured by the sum of individual satisfactions) and egalitarianism (measured by the minimum satisfaction). The quality of this tradeoff can be assessed by a convex combination of the two criteria *min* and *sum*. In cases involving more than two agents, the distribution of satisfaction over the agents can be better described by considering not only the two initial criteria *min* and *sum* but also all partial sums of type $L_k(u(\pi)) = u^{\uparrow}_1(\pi) + \ldots + u^{\uparrow}_k(\pi)$ measuring the overall satisfaction of the k least satisfied agents, $k = 1, \ldots, n$, where $u^{\uparrow}(\pi) = (u^{\uparrow}_1(\pi), \ldots, u^{\uparrow}_n(\pi))$ is the vector derived from $u(\pi)$ by sorting its components by increasing order. The vector $L(u(\pi)) = (L_1(u(\pi), \ldots, L_n(u(\pi)))$ is well known in majorization theory for inequality measurement [24]. It is also known in Social Choice theory [32] as the *generalized Lorenz vector* and used to define the *L-dominance*, a measure of relative inequalities on vectors having possibly different means. L-dominance is a strict preference relation defined on utility vectors by: $u' \succ_L u$ if $L_k(u') \geq L_k(u)$ for $k = 1 \ldots n$, one of these inequalities being strict (in other words this is the Pareto dominance over generalized Lorenz vectors).

The relationship between L-dominance and fairness can be explained as follows: let u be the utility vector of a feasible solution such that $u_i > u_j$ for two agents i, j in a multiagent problem, then for any other feasible utility vector $u' = (u_1, \ldots, u_{j-1}, u_j + \varepsilon, u_{j+1}, \ldots, u_{i-1}, u_i - \varepsilon, u_{i+1}, \ldots, u_n)$ with $0 < \varepsilon < u_i - u_j$, we have $u' \succ_L u$. We can see that the transformation (ϵ-transfer) allowing to move from u to u' reduces inequalities and improves the solution in terms of Lorenz dominance. To be more precise, it has been shown that $u' \succ_L u$ if and only if there exists a sequence of ϵ-transfers and Pareto-improvements allowing to pass from u to u' [18,7]. For example $u' = (12, 12, 11, 11, 7)$ L-dominates $u = (20, 5, 6, 11, 8)$ due to the following improving sequence: $(20, 5, 6, 11, 8) \prec (20, 5, 7, 11, 7) \prec (16, 5, 11, 11, 7) \prec (12, 9, 11, 11, 7) \prec (12, 12, 11, 11, 7)$. We have used three ϵ-transfers and one Pareto-improvement but there is no need to find such a sequence explicitly. In view of [18,7], the fact that $L(u') = (7, 18, 29, 41, 53)$ Pareto dominates $L(u) = (5, 11, 19, 30, 50)$ implies the existence of an improving sequence from u to u'. The notion of L-dominance is a refinement of Pareto dominance. Lorenz non-dominated solutions are those Pareto-optimal solutions that cannot be improved in terms of ϵ-transfer. In this respect, they can be considered as "optimally fair".

However, Lorenz dominance is a partial order and many solutions are incomparable. In the initial example we have $L(u(\pi)) = (5, 12, 23, 34, 54)$ whereas $L(u(\pi')) = (7, 18, 29, 41, 53)$. Hence none of the two vectors $u(\pi)$ and $u(\pi')$ L-dominates the other. A linear extension of L-dominance is obtained by measuring the social welfare as a weighted sum of the components of the Lorenz vector, for some positive coefficients. Applied to our framework, it amounts to defining a social welfare function as: $\psi(u(\pi)) = \sum_{k=1}^{n} \alpha_k L_k(u(\pi))$ with $\alpha_k \geq 0$ thus generalizing the basic tradeoff between utilitarism and egalitarism. Now, by setting $w_i = \sum_{k=i}^{n} \alpha_k$ and $w = (w_1, \ldots, w_n)$, ψ can be rewritten using w as follows:

$$\psi_w(u(\pi)) = \sum_{i=1}^{n} w_i u^{\uparrow}_i(\pi) \quad (2)$$

This aggregation function is an ordered weighted average (OWA) as defined in Yager [36], applied to vector $u(\pi)$. Note that, by construction, the weights are such that $w_i \geq w_{i+1}$ for all $i < n$. This is a direct consequence of the non-negativity of $\alpha_k, k = 1, \ldots, n$, a necessary condition to be monotonically increasing with respect to Lorenz dominance and therefore, with ϵ-transfers. Thus the maximization of ψ_w will favor fair solutions. For this reason, such an OWA function constructed with non-decreasing weights will be referred to as a *fair OWA* in the sequel. Coming back to the initial example, if $\alpha_k = 1$ for $k = 1, \ldots, 5$, we obtain $w_i = 6 - i$ and $\psi_w(u(\pi)) = 115$ whereas $\psi_w(u(\pi')) = 148$ which means that π' is better than π. Actually, assignment π' is ψ_w-optimal for this vector w.

Social utility functions defined with ψ_w are known as *generalized Gini social-evaluation functions* in Social Choice theory and their axiomatic justification in terms of fairness is well established [35, 32]. The aim of this paper is to study the problem of determining an assignment maximizing such a fair OWA function.

In Section 2 we recall some known results about the optimization of fair OWA functions and the complexity of some particular instances of assignment problems. Section 3 addresses the issue of the complexity status of the problem, by providing a NP-hardness proof. Section 4 is devoted to the development of an efficient algorithm for the subclass of problems with weights featuring at most k distinct values. Some conclusions and perspectives for future research are given in Section 5.

2 The fair OWA assignment problem

Let us consider an assignment problem with n agents and n objects. The fair OWA assignment problem is an optimization problem and the corresponding decision problem can be stated as follows:

> FAIR OWA MULTIAGENT ASSIGNMENT PROBLEM (FOWAMA)
> INSTANCE: A pair (U, w), where U is an integer-valued square matrix of size n, w is a vector of size n of non-negative and non-increasing weights (i.e., $w_i \geq w_{i+1}, i = 1, \ldots, n-1$), and a positive rational number Q.
> QUESTION: Does there exist an assignment (i.e., a one-to-one mapping π) such that $\psi_w(u(\pi)) \geq Q$, ψ_w being defined as in Equation (2)?

In the above problem, the vector of weights w is part of the instance. Various polynomial cases have been exhibited in the literature. For example, when $w = (1, \ldots, 1)$, i.e., when the sum of individual utilities is to be maximized, the well known Hungarian algorithm [21] solves the problem in polynomial time. Also, when $w = (1, 0, \ldots, 0)$, OWA boils down to the minimum and this corresponds to bottleneck optimization, a well-known polynomial case [11]. The linear combination of bottleneck optimization and max-sum optimization corresponding to weights of type $w = (p, 1, \ldots, 1)$ with $p > 1$ is also solvable in polynomial time [25]. When $w = (1, \ldots 1, 0, \ldots, 0)$, OWA reduces to a single Lorenz component L_k and the problem is known as the k-sum assignment problem; it can be solved in polynomial time [17]. When all the differences $w_i - w_{i+1}$ are sufficiently large, OWA optimization is equivalent to leximin optimization, another polynomial case solved in Sokkalingam et al. [33]. Finally, if we relax the non-negativity constraints on the weights, another case corresponding to $w = (1, 0, \ldots, 0, -1)$ has been considered in Gupta et al. [16] to solve the so-called minimum-deviation problem; here also, the problem is proved to be polynomial.

Despite the fact that various polynomial cases are known, the complexity of the fair OWA multiagent problem has been until now an open question. There are NP-hardness results for the one-to-many OWA maximum assignment problem version, see e.g. [4,13] but the proofs do not apply to the one-to-one assignment problem. Finally, nice linearizations of the OWA operator have been proposed to reformulate OWA optimization problems in the language of linear programming [29,2,5]. However, in all cases, applying these techniques on the fair OWA multi-agent assignment problem requires to solve a mixed integer program, which does not tell us anything on the complexity of the problem. In the next section we prove that the FOWAMA problem is NP-complete.

3 Complexity of the FOWAMA problem

In order to prove that the FOWAMA problem is NP-complete we describe a polynomial-time many-one reduction (a.k.a. Karp reduction) of MAX3SAT to FOWAMA, the former problem being defined as follows (see [10]):

Maximum 3-Satisfiability (MAX3SAT)
instance: A set $X = \{x_1, \ldots, x_p\}$ of p boolean variables, a collection $C = \{C_1, \ldots, C_m\}$ of clauses over X such that each clause $C_i \in C$ has 3 literals, and a positive integer $R \leq m$.
question: Is there a truth assignment for X that simultaneously satisfies at least R clauses in C?

Because solving any instance of the 3-SAT problem (which is NP-complete [8]) is equivalent to deciding whether the corresponding MAX3SAT instance with $R = m$ (m being the number of clauses) admits a truth assignment satisfying all clauses, MAX3SAT is also NP-complete. We now introduce a reduction from MAX3SAT to FOWAMA.

3.1 Reduction from MAX3SAT to FOWAMA

Let $M = \{1, \ldots, m\}$ and $P = \{1, \ldots, p\}$ be the set of indices for clauses and variables, respectively. In order to reduce the MAX3SAT problem to the FOWAMA problem, we are going to associate to any instance of MAX3SAT a bipartite graph $G = (V, E)$, where V is a set of $12m$ vertices partitioned into $6m$ agent-vertices and $6m$ object-vertices and E is a subset of edges to be defined below.

For any variable x_i, let s_i^+ (respectively, s_i^-) denote the number of occurrences of literals x_i (respectively, $\neg x_i$) in C, and S_i^+ (respectively, S_i^-) denote the ordered list of clause indices where x_i (respectively, $\neg x_i$) appears. We can assume without loss of generality that $s_i^+ > 0$ and $s_i^- > 0$ for any $i \in P$. Furthermore, let $s_i = s_i^+ + s_i^-$ and S_i be the ordered list resulting from the union of S_i^+ and S_i^-. For any variable x_i we define s_i "positive" agent-vertices denoted $a_i^j, j \in S_i$ and s_i "negative" agent-vertices denoted $\bar{a}_i^j, j \in S_i$. Overall we obtain $2\sum_{i \in M} s_i = 6m$ agent-vertices. Moreover, for any clause $C_j, j \in M$ we define three "true" object-vertices respectively denoted t_j^*, t_j^1, t_j^2 and 3 "false" object-vertices respectively denoted f_j^1, f_j^2, f_j^3. Hence, we also obtain a set of $6m$ object-vertices. Assuming that the three variables occurring in C_j are x_i, x_k, x_l, we denote G_j the subgraph of G reduced to vertices associated with clause C_j defined in Figure 1.

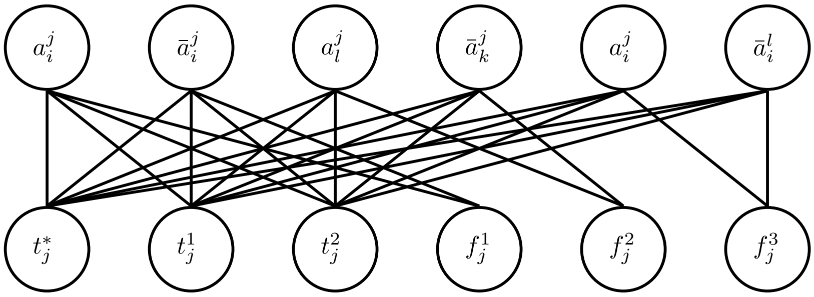

Fig. 1 The edges with nonzero utilities in G_j

The set of agents-vertices of G_j is $A_j = \{a_i^j, \bar{a}_i^j, a_k^j, \bar{a}_k^j, a_l^j, \bar{a}_l^j, \}$ and the set of object-vertices of G_j is $O_j = \{t_j^*, t_j^1, t_j^2, f_j^1, f_j^2, f_j^3\}$. Thus, G_j is an assignment graph of objects in O_j to agents in A_j. Let $A = \bigcup_{j \in M} A_j$ and $O = \bigcup_{j \in M} O_j$.

We will show later that by setting to zero the utilities of edges in $A_i \times O_l$ for any $l \neq i$, the overall assignment of objects in O to agents in A amounts to solving a series of m interdependent assignment problems on pairs $(A_j, O_j), j = 1, \dots m$.

Let us explain now the connection between solutions to the assignment problem on the graph G_j and truth assignments of variables x_i, x_k, x_l occurring in clause C_j in this order. Here also, some edges are irrelevant to model truth assignments and are penalized in order to be excluded from any optimal assignment. This is done by assigning zero utility value to such edges. The edge set of G_j is constructed as follows: each of the true vertices (t_j^*, t_j^1, t_j^2) is connected to all agents-vertices in A_j; false vertex f_j^1 is connected to two agent-vertices a_i^j and $\bar{a}_i^j$, false vertex f_j^2 is connected to two agent-vertices a_i^k and $\bar{a}_i^k$, false vertex f_j^2 is connected to two agent-vertices a_i^l and $\bar{a}_i^l$. Figure 1 shows the only edges with nonzero utility in G_j.

Any solution of the 6×6 assignment problem on G_j (Figure 1) corresponds to a truth assignment of variables x_i, x_k, x_l satisfying clause C_j. The truth value of variable x_i is derived as follows: if object f_j^1 is assigned to agent a_i^j (resp. $\bar{a}_i^j$) then $x_i = false$ (resp. $x_i = true$). A similar rule applies for x_k (resp. x_l) using object f_j^2 (resp. f_j^3). Note that there may exist several assignments representing the same truth values for x_i, x_k, x_l due to the freedom left in the choice of true-objects assigned to agents.

Now, if we consider all clauses simultaneously, it may be the case that partial assignments determined independently on the various subgraphs $G_j, j = 1, \dots, m$ may not correspond to a consistent truth-assignment of all variables. A variable x_i may be true in clause G_j but false in clause C_k. We are going to show that there exists a specific choice of utilities and OWA weights ensuring the overall consistency of truth-assignments corresponding to the optimal solutions of the FOWAMA instance, whenever the MAX3SAT instance is satisfiable. We explain below the construction of such utilities and weights for any arbitrary instance of MAX3SAT.

Definition of utilities and OWA weights. First, let us mention that the utility of the edges in E linking two vertices of two distinct subgraphs $G_j, G_k, j \neq k$ is set to 0. This is used to penalize any assignment of an object of O_k to an agent of A_j. The utility values for the other edges are chosen sufficiently large to ensure that no OWA-optimal assignment will use one of these edges, as will be shown later.

We now define the utilities of the edges of subgraphs $G_j, j \in M$. Recall that an edge represents the possibility of matching an object to an agent. Therefore, the utility of an edge represents the utility of the object for the agent under consideration. If clause C_j includes variables x_i, x_k, x_l in this order, $i \neq 1, j \neq 1, l \neq 1$, then the utilities of the edges linking a_i^j and $\bar{a}_i^j$ (respectively, a_k^j and $\bar{a}_k^j$, a_l^j and $\bar{a}_l^j$) to $t_j^*, t_j^1, t_j^2, f_j^1, f_j^2, f_j^3$ are given in Tables 1–3. If one of the indices i, j, k is equal to 1, the corresponding table is modified by replacing its first line with the one given in Table 4.

Table 1 Utilities of the edges involving agent-vertices associated with x_i

x_i	a_i^j	$\bar{a}_i^j$
t_j^*	$2^{6im+4m} + 2^{6im+5m}$ if $j \in S_i^+$ 2^{6im+4m} if $j \in S_i^-$	$2^{6im+m} + 2^{6im+2m}$ if $j \in S_i^-$ 2^{6im+m} if $j \in S_i^+$
t_j^1	2^{6im+4m}	2^{6im+m}
t_j^2	2^{6im+4m}	2^{6im+m}
f_j^1	2^{6im+3m}	2^{6im}
f_j^2	0	0
f_j^3	0	0

Table 2 Utilities of the edges involving agent-vertices associated with x_k

x_k	a_k^j	$\bar{a}_k^j$
t_j^*	$2^{6km+4m} + 2^{6km+5m}$ if $j \in S_i^+$ 2^{6km+4m} if $j \in S_i^-$	$2^{6km+m} + 2^{6km+2m}$ if $j \in S_i^-$ 2^{6km+m} if $j \in S_i^+$
t_j^1	2^{6km+4m}	2^{6km+m}
t_j^2	2^{6km+4m}	2^{6km+m}
f_j^1	0	0
f_j^2	2^{6km+3m}	2^{6km}
f_j^3	0	0

Note that, in Tables 1–3, some utilities are set to 0. This is to penalize the edges that do not appear in Figure 1; they are indeed irrelevant to interpret an assignment in G as a truth-assignment.

Let us explain the range of possible non-zero utility values representing the satisfactions of agents $\{a_1^j, \bar{a}_1^j\}, j \in S_1$ (agents concerned with the truth assignment of variable x_1). For agent $\bar{a}_1^j$, non-zero utilities range from 2^{6m} to $2^{6m+m} + 2^{2m}$.

Table 3 Utilities of the edges involving agent-vertices associated with x_l

x_l	a_l^j	$\bar{a}_l^j$
t_j^*	$2^{6lm+4m} + 2^{6lm+5m}$ if $j \in S_i^+$ 2^{6lm+4m} if $j \in S_i^-$	$2^{6lm+m} + 2^{6lm+2m}$ if $j \in S_i^-$ 2^{6lm+m} if $j \in S_i^+$
t_j^1	2^{6lm+4m}	2^{6lm+m}
t_j^2	2^{6lm+4m}	2^{6lm+m}
f_j^1	0	0
f_j^2	0	0
f_j^3	2^{6lm+3m}	2^{6lm}

Table 4 Utilities of edges involving agent-vertices associated with variable x_1

x_1	a_1^j	$\bar{a}_1^j$
t_j^*	$2^{6m+4m} + 2^{5m}$ if $j \in S_1^+$ 2^{6m+4m} if $j \in S_1^-$	$2^{6m+m} + 2^{2m}$ if $j \in S_1^-$ 2^{6m+m} if $j \in S_1^+$

The lower utility value follows from assigning the false object f_j^k to agent $\bar{a}_1^j$ where x_1 (or $\neg x_1$) is the k-th literal of C_j. The higher utility value follows from assigning the true object t_j^* to agent $\bar{a}_1^j$ where x_1 (or $\neg x_1$) is a literal of C_j. For agent a_1^j, non-zero utilities range from 2^{6m+3m} to $2^{6m+4m} + 2^{5m}$. The lower utility value follows from assigning the false object f_j^k to agent a_1^j where x_1 (or $\neg x_1$) is the k-th literal of C_j. The higher utility value follows from assigning the true object t_j^* to agent a_1^j where x_1 (or $\neg x_1$) is a literal of C_j. Similarly the range of possible non-zero utility can be defined for the agents concerned with the truth assignment of the other variables $x_i, i > 1$. Table 5 summarizes the ranges of possible non-zero utilities for the different groups of agents:

Table 5 Ranges of non-zero utility values for all groups of agents

agent	range of non-zero utility
$\bar{a}_1^j$	$[2^{6m},\ 2^{6m+m} + 2^{2m}]$
a_1^j	$[2^{9m},\ 2^{6m+4m} + 2^{5m}]$
$\bar{a}_i^j, i > 1$	$[2^{6im},\ 2^{6im+m} + 2^{6im+2m}]$
$a_i^j, i > 1$	$[2^{6im+3m},\ 2^{6im+4m} + 2^{6im+5m}]$

Remark 1 *From the above construction, it is seen that two intervals of possible utility values corresponding to two distinct variables never overlap. Hence, for any assignment π providing all agents with a strictly positive utility, the components of the sorted utility vector $u^{\uparrow}(\pi)$ necessarily appear according to the following sequence:*

$$\left(\{\bar{a}_1^j, j \in S_1\}, \{a_1^j, j \in S_1\}, \ldots, \{\bar{a}_p^j, j \in S_p\}, \{a_p^j, j \in S_p\}\right). \quad (3)$$

Let $\bar{\alpha}_k = 1 + 2\sum_{i=1}^{k-1} s_i$ be the smallest index of a term of type $\bar{a}_k^j$ in the sequence (3) and $\alpha_k = \bar{\alpha}_k + s_k = 1 + s_k + 2\sum_{i=1}^{k-1} s_i$ the smallest rank of a term of type a_k^j. These indices will be useful to determine the smallest individual utility obtained within specific groups of agents. For example $u^{\uparrow}_{\bar{\alpha}_i}(\pi)$ (resp. $u^{\uparrow}_{\alpha_i}(\pi)$) represents the smallest utility obtained within the group $\{\bar{a}_i^j, j \in S_i\}$ (resp. $\{a_i^j, j \in S_i\}$).

Finally we specify the weights w_k that will be used to define the OWA aggregation functions: $\psi_w(u(\pi)) = \sum_{k=1}^{6m} w_k u_k^{\uparrow}(\pi)$:

$$w_k = \begin{cases} 2^{3m} & \text{if } k = \bar{\alpha}_1, \\ 2^0 & \text{if } k \in [\bar{\alpha}_1 + 1, \alpha_1], \\ 2^{-3m} & \text{if } k \in [\alpha_1 + 1, \alpha_1 + s_1 - 1], \\ 2^{3m-6im} & \text{if } k = \bar{\alpha}_i, i \in \{2, \ldots, p\}, \\ 2^{-6im} & \text{if } k \in [\bar{\alpha}_i + 1, \alpha_i], i \in \{2, \ldots, p\}, \\ 2^{-3m-6im} & \text{if } k \in [\alpha_i + 1, \alpha_i + s_i - 1], i \in \{2, \ldots, p\}. \end{cases} \tag{4}$$

Note that these weights decrease as k increases from 1 to $6m$.

3.2 Preliminary results

Let us introduce the concept of *consistent* assignment in G, that can be associated to a truth-assignment of variables in MAX3SAT.

Definition 1 *An assignment π in G is said to be* consistent *if for any $i \in P$ one of the two following cases occurs :*

i) for all $j \in S_i$ agent a_i^j is matched with an object belonging to $\{t_j^*, t_j^1, t_j^2\}$ and $\bar{a}_i^j$ is matched with the object f_j^l, l being the position (1, 2, 3) of variable x_i in clause C_j,

ii) for all $j \in S_i$ agent $\bar{a}_i^j$ is matched with an object belonging to $\{t_j^*, t_j^1, t_j^2\}$ and a_i^j is matched with the object f_j^l, l being the position (1, 2, 3) of variable x_i in clause C_j.

The first case (i) corresponds to the assignment $x_i = \mathit{true}$ in all clauses C_j involving an occurrence of x_i. Similarly, the second case (ii) corresponds to the assignment $x_i = \mathit{false}$ in all clauses C_j involving an occurrence of x_i. So, it is possible to associate with any consistent assignment π, a truth assignment x^π of the boolean variables X such that for any $i \in P$, $x_i^\pi = \mathit{true}$ iff for all $j \in S_i$ agent $\bar{a}_i^j$ is matched with an object belonging to $\{t_j^*, t_j^1, t_j^2\}$ (case (ii) of Definition 1). For any $i \in P$ such that $x_i^\pi = \mathit{true}$ (resp. $x_i^\pi = \mathit{false}$), let q_i^π denote the number of agents a_i^j (resp. $\bar{a}_i^j$), with $j \in S_i^+$ (resp. $j \in S_i^-$) matched with an object t_j^* in assignment π, and let $q^\pi = \sum_{i \in P} q_i^\pi$. Note that for any $i \in P$ such that $x_i^\pi = \mathit{true}$ we have $q_i^\pi \leq s_i^+$ and therefore, $q_i^\pi < s_i^+ + s_i^- = s_i$ since $s_i^- > 0$. Similarly, for any $i \in P$ such that $x_i^\pi = \mathit{false}$ we have $q_i^\pi \leq s_i^-$ and therefore, $q_i^\pi < s_i^+ + s_i^- = s_i$ since $s_i^+ > 0$. Hence, in both cases we have $q_i^\pi < s_i$.

We now establish the following Lemma:

Lemma 1 *Any consistent assignment π is such that $\psi_w(u(\pi)) = K + 2^{2m} q^\pi$ with*

$$K = 2^{10m} + 2^{9m} + (s_1 - 1)(2^{7m} + 2^{6m}) + (p-1)(2^{4m} + 2^{3m}) + (3m - p - s_1 + 1)(2^m + 1). \tag{5}$$

Proof Let z be the utility vector of the consistent assignment π. Note that all components of z are strictly positive since no edge with utility zero belongs to a consistent assignment, by definition. Due to Remark 1, the s_1 first components of $z^{\uparrow}$ correspond to the satisfaction of agents in $\{\bar{a}_1^j, j \in S_1\}$ and the s_1 next components correspond to the satisfaction of agents in $\{a_1^j, j \in S_1\}$. If the consistent assignment under consideration corresponds to $x_1^\pi = true$ (case *(i)* of Definition 1) then $z_k^{\uparrow} = 2^{6m}$ for $k \in [1, s_1]$, $z_k^{\uparrow} = 2^{10m}$ for $k \in [s_1+1, 2s_1 - q_1^\pi]$, and $z_k^{\uparrow} = 2^{10m} + 2^{5m}$ for $k \in [2s_1 - q_1^\pi + 1, 2s_1]$. Hence, $\sum_{k=1}^{2s_1} w_k z_k^{\uparrow} = \sum_{k=1}^{s_1} w_k z_k^{\uparrow} + \sum_{k=s_1+1}^{2s_1-q_1^\pi} w_k z_k^{\uparrow} + \sum_{k=2s_1-q_1^\pi+1}^{2s_1} w_k z_k^{\uparrow} = 2^{6m}(2^{3m} + s_1 - 1) + 2^{10m}(1 + 2^{-3m}(s_1 - q_1^\pi - 1)) + 2^{-3m} q_1^\pi (2^{10m} + 2^{5m}) = 2^{9m} + 2^{10m} + (s_1 - 1)(2^{6m} + 2^{7m}) + 2^{2m} q_1^\pi = 2^{6m}(2^{3m} + 2^{4m} + (s_1 - 1)(1 + 2^m)) + 2^{2m} q_1^\pi$.

Similarly, if the assignment under consideration corresponds to $x_1^\pi = false$ (case *(ii)* of Definition 1) then $z_k^{\uparrow} = 2^{7m}$ for $k \in [1, s_1 - q_1^\pi]$, $z_k^{\uparrow} = 2^{7m} + 2^{2m}$ for $k \in [s_1 - q_1^\pi + 1, s_1]$, and $z_k^{\uparrow} = 2^{9m}$ for $k \in [s_1 + 1, 2s_1]$. Hence, $\sum_{k=1}^{2s_1} w_k z_k^{\uparrow} = \sum_{k=1}^{s_1-q_1^\pi} w_k z_k^{\uparrow} + \sum_{k=s_1-q_1^\pi+1}^{s_1} w_k z_k^{\uparrow} + \sum_{k=s_1+1}^{2s_1} w_k z_k^{\uparrow} = 2^{7m}(2^{3m} + s_1 - q_i^\pi - 1) + q_1^\pi(2^{7m} + 2^{2m}) + 2^{9m}(1 + 2^{-3m}(s_1 - 1)) = 2^{9m} + 2^{10m} + (s_1 - 1)(2^{6m} + 2^{7m}) + 2^{2m} q_1^\pi = 2^{6m}(2^{3m} + 2^{4m} + (s_1 - 1)(1 + 2^m)) + 2^{2m} q_1^\pi$.

Thus, whatever the truth value assigned to x_1^π, we obtain:

$$\sum_{k=1}^{2s_1} w_k z_k^{\uparrow} = 2^{6m}(2^{3m} + 2^{4m} + (s_1 - 1)(1 + 2^m)) + 2^{2m} q_1^\pi. \tag{6}$$

Using similar arguments, it can be shown that for $i > 1$ we have:

$$\sum_{k=\bar{\alpha}_i}^{\alpha_i+s_i-1} w_k z_k^{\uparrow} = 2^{3m} + 2^{4m} + (s_i - 1)(1 + 2^m) + 2^{2m} q_i^\pi. \tag{7}$$

Hence, $\psi_w(z) = \sum_{k=1}^{6m} w_k z_k^{\uparrow} = \sum_{k=1}^{2s_1} w_k z_k^{\uparrow} + \sum_{i=2}^{p} \sum_{k=\bar{\alpha}_i}^{\alpha_i+s_i-1} w_k z_k^{\uparrow} = (2^{3m} + 2^{4m})(2^{6m} + (p-1)) + (1 + 2^m)(2^{6m}(s_1 - 1) + \sum_{i=2}^{p}(s_i - 1)) + \sum_{i \in P} 2^{2m} q_i^\pi = (2^{3m} + 2^{4m})(2^{6m} + (p-1)) + (1 + 2^m)(2^{6m}(s_1 - 1) + (3m - s_1 - p + 1)) + 2^{2m} \sum_{i \in P} q_i^\pi = K + 2^{2m} q^\pi$. □

Corollary 1 *Any consistent assignment π is such that $\psi_w(u(\pi)) \geq K$.*

We want to prove now that a OWA-optimal assignment of objects to agents only uses edges with non-zero utility. To this end we make the following preliminary remarks:

Remark 2 *For any assignment π, for any index $i \in P$, the least satisfied agent in $\{a_i^j, j \in S_i\}$ receives an object with utility not larger than 2^{6im+4m}, and the least satisfied agent in $\{\bar{a}_i^j, j \in S_i\}$ receives an object with utility not larger than 2^{6im+m}. Let $j \in S_i^+$ and $k \in S_i^-$. Then the maximal possible utility obtained for agent $\bar{a}_i^j$ (resp. a_i^k) results from the assignment of object t_j^* (resp. t_k^*), see Tables 1–4, and is equal to 2^{6im+m} (resp. 2^{6im+4m}).*

Remark 3 *We recall that the set of clauses C admits at least one positive and one negative occurrence of x_i (otherwise the variable can be instanciated and eliminated). Therefore, there exist at least 2p literals in C; hence, $m \geq 2p/3$ and therefore, $p \leq 3m/2$.*

Lemma 2 *For any assignment π, and for any index $k = 1 \dots 6m$, the inequality $u^{\uparrow}_k(\pi) \leq b_k$ holds, where b_k is defined as follows.*

$$b_k = \begin{cases} 2^{7m} & \text{if } k = \bar{\alpha}_1, \\ 2^{7m} + 2^{2m} & \text{if } k \in [\bar{\alpha}_1 + 1, \alpha_1 - 1], \\ 2^{10m} & \text{if } k = \alpha_1, \\ 2^{6m+4m} + 2^{5m} & \text{if } k \in [\alpha_1 + 1, \alpha_1 + s_1 - 1], \\ 2^{6im+m} & \text{if } k = \bar{\alpha}_i, i \in \{2, \dots, p\}, \\ 2^{6im+m} + 2^{6im+2m} & \text{if } k \in [\bar{\alpha}_i + 1, \bar{\alpha}_i + s_i - 1], i \in \{2, \dots, p\}, \\ 2^{6im+4m} & \text{if } k = \alpha_i, i \in \{2, \dots, p\}, \\ 2^{6im+4m} + 2^{6im+5m} & \text{if } k \in [\alpha_i + 1, \alpha_i + s_i - 1], i \in \{2, \dots, p\}. \end{cases} \tag{8}$$

Proof From Remark 2, we know that the lowest utility of an agent belonging to $\{\bar{a}^j_1, j \in S_1\}$ is bounded above by 2^{6m+m}. So we can state that $u^{\uparrow}_{\bar{\alpha}_1}(\pi) \leq 2^{7m}$. We know that utilities of the agents belonging to $\{\bar{a}^j_1, j \in S_1\}$ are bounded above by $2^{6m+m} + 2^{2m}$ (see Table 4). So we have $u^{\uparrow}_k(\pi) \leq 2^{7m} + 2^{2m}$ for any $k \in [\bar{\alpha}_1 + 1, \alpha_1 - 1]$. From Remark 2, we know that the lowest utility of an agent belonging to $\{a^j_1, j \in S_1\}$ is bounded above by 2^{6m+4m}. So we can state that $u^{\uparrow}_{\alpha_1}(\pi) \leq 2^{10m}$. We also know that utilities of the agents belonging to $\{a^j_1, j \in S_1\}$ are bounded above by $2^{6m+4m}+2^{5m}$ (see Table 5). So we have $u^{\uparrow}_k(\pi) \leq 2^{10m}+2^{5m}$ for all $k \in [\alpha_1 + 1, \alpha_1 + s_1 - 1]$. More generally we know from Remark 2 that for all $i \in P$ we have $u^{\uparrow}_{\bar{\alpha}_i} \leq 2^{6im+m}$ and $u^{\uparrow}_{\alpha_i} \leq 2^{6im+4m}$. Moreover, we have $u^{\uparrow}_k \leq 2^{6im+m}+2^{6im+2m}$ for all $k \in [\bar{\alpha}_i+1, \bar{\alpha}_i+s_i-1]$ and $u^{\uparrow}_l \leq 2^{6im+4m}+2^{6im+5m}$ for all $l \in [\alpha_i + 1, \alpha_i + s_i - 1]$.

Lemma 3 *Any assignment π such that at least one agent is matched with a zero utility object is such that $\psi_w(u(\pi)) < K$, K being defined by Equation 5.*

Proof Let π be an assignment whose utility vector $u(\pi)$ contains at least one component 0. We first determine an upper bound on $\psi_w(u(\pi))$. By hypothesis, we have $u^{\uparrow}_1(\pi) = 0$ because the utility values are all non-negative. Furthermore, by Lemma 2 we have $u^{\uparrow}_k(\pi) \leq b_k$ for all $k = 2, \dots, 6m$. Hence, using weights defined in Equation (4), we obtain :

$$\begin{aligned} \psi_w(u(\pi)) \leq {} & (s_1 - 1)(2^{7m} + 2^{2m}) + 2^{10m} + (s_1 - 1)(2^{7m} + 2^{2m}) \\ & +2 \cdot 2^{4m}(p - 1) + (6m - 2s_1 - 2p + 2)(2^{2m} + 2^m). \end{aligned} \tag{9}$$

The first three terms in (9) are straightforward. The term $2 \cdot 2^{4m}(p-1)$ comes from the $2(p-1)$ terms of the form $2^{-6im+3m} u^{\uparrow}_{\bar{\alpha}_i}(\pi)$ or $2^{-6im} u^{\uparrow}_{\alpha_i}(\pi)$ with $i \in N \backslash \{1\}$ (we replace the values $u^{\uparrow}_{\bar{\alpha}_i}(\pi)$ and $u^{\uparrow}_{\alpha_i}(\pi)$ by $b_{\bar{\alpha}_i}$ and b_{α_i} respectively; we obtain $2^{-6im+3m} u^{\uparrow}_{\bar{\alpha}_i}(\pi) = 2^{-6im} u^{\uparrow}_{\alpha_i}(\pi) = 2^{4m}$). Finally the term $(6m - 2s_1 - 2p + 2)(2^{2m} + 2^m)$ comes from the remaining $(6m - 2s_1 - 2p + 2)$ terms, either of the form $2^{-6im} u^{\uparrow}_k(\pi)$ for $k \in [\bar{\alpha}_i + 1, \bar{\alpha}_i + s_i - 1], i \in \{2, \dots, p\}$, or of the form $2^{-6(i+1)m+3m} u^{\uparrow}_l(\pi)$ for $l \in [\alpha_i + 1, \alpha_i + s_i - 1], i \in \{2, \dots, p\}$, each of them being equal to $2^{2m} + 2^m$.

Finally, to prove that $\psi_w(u(\pi)) < K$ let us consider the quantity: $K' = (s_1 - 1)2^{7m} + 2^{10m} + 2^{4m}(p - 1)$. Using Equation (5) we have:

$$K' < K - 2^{9m}. \tag{10}$$

Moreover, by Equation (9) we obtain:

$$\psi_w(u(\pi))-K' \leq (s_1-1)(2^{7m}+2^{2m+1})+2^{4m}(p-1)+(6m-2s_1-2p+2)(2^{2m}+2^m).$$

By Remark 3 we know that $p \leq 3m/2 \leq 2m$. Moreover, we have $s_1 \leq m$. Finally, we assume that $m \geq 4$ (to simplify the proof, smaller instances are irrelevant in view of our complexity analysis). Under these conditions, we obtain the following inequalities:

$$\begin{aligned}(s_1-1)(2^{7m}+2^{2m+1})+2^{4m}(p-1)+(6m-2s_1-2p+2)(2^{2m}+2^m)\\ < m(2^{7m+1}+2^{4m+1}+6\cdot 2^{2m+1}) < m(2^{7m+2}+2^{2m+4})\\ < m2^{7m+3} < 2^m \cdot 2^{8m} = 2^{9m}.\end{aligned}$$

Therefore, we get:

$$\psi_w(u(\pi)) - K' < 2^{9m}. \tag{11}$$

Hence, by adding Equations (10) and (11) we obtain $\psi_w(u(\pi)) < K$. □

We have shown in Corollary 1 that any consistent assignment π is such that $\psi_w(u(\pi)) \geq K$. One can construct a consistent assignment by assigning items to agents according to *(i)* of Definition 1. This consistent assignment corresponds to the truth-assignment where all variables are set to true. Since a consistent assignment exists for any instance, we deduce that any optimal assignment is such that $\psi_w(u(\pi)) \geq K$. Hence, Lemma 3 shows that any assignment providing at least one agent with a zero utility object is suboptimal. Therefore, any optimal assignment π leads to a utility vector $u(\pi)$ with strictly positive components. Let us introduce now a fourth Lemma:

Lemma 4 *Any assignment π yielding a strictly positive utility vector $u(\pi)$ and such that $\psi_w(u(\pi)) \geq K$ is consistent.*

Proof By contraposition we want to prove that, for any assignment π yielding a strictly positive utility vector $u(\pi)$, if π is not consistent then $\psi_w(u(\pi)) < K$. Let π be an assignment with strictly positive utility vector $u(\pi)$. Hence, every *false* object f_j^l ($j \in M, l \in \{1,2,3\}$) is assigned either to agent a_i^j or to agent $\bar{a}_i^j$ where x_i is the variable appearing in position l in clause C_j (otherwise we would obtain a zero component in $u(\pi)$ by definition of the utility functions, see Tables 1–4 and Figure 1). Since π is not consistent, there exists at least one variable for which truth assignments are not consistent. Let i be the smallest variable index for which such an inconsistency appears. The inconsistency is characterized by the existence of two false objects f_q^l and f_t^k assigned to a_i^q and $\bar{a}_i^t$, respectively, where x_i is a variable appearing both in position l in clause C_q and in position k in clause C_t. From Tables 1–3, we know that the utility of f_t^k is 2^{6im} for agent $\bar{a}_i^t$ and the utility of f_q^l is 2^{6im+3m} for agent a_i^q. Furthermore, from Table 5 we know that the range of utility for agents a_i^j with $j \in P$ is $[2^{6im+3m},\ 2^{6im+4m}+2^{6im+5m}]$. Since the values $u^{\uparrow}_{\bar{\alpha}_i}(\pi)$ and $u^{\uparrow}_{\alpha_i}(\pi)$ correspond to the lowest utility of an agent of $\{\bar{a}_i^j, j \in S_i\}$ and $\{a_i^j, j \in S_i\}$, respectively, we have $u^{\uparrow}_{\bar{\alpha}_i}(\pi) = 2^{6im}$ and $u^{\uparrow}_{\alpha_i}(\pi) = 2^{6im+3m}$.

Now, in order to find an upper bound of $\psi_w(u(\pi))$ we use the upper bounds b_k of the components of $u^{\uparrow}_k(\pi)$ introduced in Lemma 2 combined with the additional information that all components of $u(\pi)$ are strictly positive. We know that for any

$r \in P\backslash\{1\}$ the values $u^{\uparrow}_{\bar{\alpha}_r}(\pi)$ and $u^{\uparrow}_{\alpha_r}(\pi)$ correspond to the lowest utility of an agent of $\{\bar{a}^j_r, j \in S_r\}$ and $\{a^j_r, j \in S_r\}$, respectively. We have $u^{\uparrow}_{\bar{\alpha}_r}(\pi) \leq b_{\bar{\alpha}_r} = 2^{6rm+m}$ and $u^{\uparrow}_{\alpha_r}(\pi) \leq b_{\alpha_r} = 2^{6rm+4m}$. We claim that it is impossible to reach these two upper bounds with the same assignment π. Indeed to get $u^{\uparrow}_{\bar{\alpha}_r}(\pi) = 2^{6rm+m}$ and $u^{\uparrow}_{\alpha_r}(\pi) = 2^{6rm+4m}$, no false item must be assigned to an agent in $\{\bar{a}^j_r, j \in S_r\}$ nor to an agent in $\{a^j_r, j \in S_r\}$. But in that case, there exists a false object f^d_j for some $j \in S_r$, corresponding to the occurrence of x_r in C_j which would not be assigned to $\bar{a}^j_r$ nor to a^j_r. Hence, the assignment of this object would induce a zero utility which yields a contradiction with the assumption that $u(\pi)$ is strictly positive. This implies that in the best cases, we have $u^{\uparrow}_{\bar{\alpha}_r}(\pi) = 2^{6rm+m}$ and $u^{\uparrow}_{\alpha_r}(\pi) = 2^{6rm+3m}$, or $u^{\uparrow}_{\bar{\alpha}_r}(\pi) = 2^{6rm}$ and $u^{\uparrow}_{\alpha_r}(\pi) = 2^{6rm+4m}$. Thus, in both cases the value of $w_{\bar{\alpha}_r}u^{\uparrow}_{\bar{\alpha}_r}(\pi) + w_{\alpha_r}u^{\uparrow}_{\alpha_r}(\pi)$ is equal to $2^{-6rm+3m} \cdot 2^{6rm+m} + 2^{-6rm} \cdot 2^{6rm+3m} = 2^{-6rm+3m} \cdot 2^{6rm} + 2^{-6rm} \cdot 2^{6rm+4m} = 2^{3m} + 2^{4m}$. Hence we have:

$$w_{\bar{\alpha}_r}u^{\uparrow}_{\bar{\alpha}_r}(\pi) + w_{\alpha_r}u^{\uparrow}_{\alpha_r}(\pi) \leq 2^{3m} + 2^{4m} \text{ for all } r \in P \setminus \{1\}. \tag{12}$$

Moreover, using a similar reasoning for $r = 1$ we obtain:

$$w_{\bar{\alpha}_1}u^{\uparrow}_{\bar{\alpha}_1}(\pi) + w_{\alpha_1}u^{\uparrow}_{\alpha_1}(\pi) \leq 2^{9m} + 2^{10m}. \tag{13}$$

Finally, we know that for any $j \in S_1$, one of the two agents $\bar{a}^j_1$ and a^j_1 should receive a false object. So half of the agents of $\{a^j_1, \bar{a}^j_1, j \in S_1\}$ should receive a false object. If a false object is assigned to an agent in $\{\bar{a}^j_1, j \in S_1\}$ except the least satisfied of them it yields a utility of 2^{6m+3m} that will be multiplied by the weight 2^{-3m} in the computation of $\psi_w(u(\pi))$; if a false object is assigned to an agent in $\{a^j_1, j \in S_1\}$ except the least satisfied of them it yields a utility of 2^{6m} that will be multiplied by the weight 2^0 in the computation of $\psi_w(u(\pi))$. So in both cases, the weighted contribution of the agent in the OWA computation is 2^{6m}. Using this for all but two agents in S_1 and using Equation (12) and (13), and inequalities $u^{\uparrow}_k(\pi) \leq b_k$ with b_k defined as in Equation (8), we obtain the following inequality under the assumption that $i \neq 1$ (the case $i = 1$ will be treated separately at the end of the proof):

$$\begin{aligned} \psi_w(u(\pi)) \leq\ & 2^{9m} + 2^{10m} + (s_1 - 1)(2^{7m} + 2^{2m}) + (s_1 - 1)2^{6m} \\ & + (2^{3m} + 2^{4m})(p - 2) + 2 \cdot 2^{3m} \\ & + (2^{2m} + 2^m)(6m - 2s_1 - 2p + 2). \end{aligned} \tag{14}$$

In the right-hand side of (14), the term $2^{9m} + 2^{10m}$ directly follows from (13). Since $i \neq 1$ there is no inconsistency for variable x_1. Therefore, either all agents of the set $\{\bar{a}^j_1, j \in S_1\}$ or all agents of the set $\{a^j_1, j \in S_1\}$ receive a false object, thus inducing a contribution to $\psi_w(u(\pi))$ at least equal to $2^6m(s_1 - 1)$ (there are s_1 such agents but the contribution of one of them is already included in the term $2^{9m} + 2^{10m}$). For the remaining agents in $\{\bar{a}^j_1, j \in S_1\} \cup \{a^j_1, j \in S_1\}$, we use inequality $u^{\uparrow}_k(\pi) \leq b_k$ and Equation (8) which yields a contribution of $(s_1 - 1)2^{7m} + 2^{2m}$. This completes the arguments concerning the contribution of the agents belonging to $\{\bar{a}^j_1, j \in S_1\}$ and those belonging to $\{a^j_1, j \in S_1\}$. Let us consider now the satisfaction of agents in $\{\bar{a}^j_i, j \in S_i\}$ or in $\{a^j_i, j \in S_i\}$. We have $w_{\bar{\alpha}_i}u^{\uparrow}_{\bar{\alpha}_i}(\pi) + w_{\alpha_i}u^{\uparrow}_{\alpha_i}(\pi) = 2^{3m-6im} \cdot 2^{6im} + 2^{-6im} \cdot 2^{6im+3m} = 2 \cdot 2^{3m}$ corresponding to the contribution of the least satisfied pair of agent (one in each of these two sets). Then, using Equation (12) for all $r \in P \setminus \{1, i\}$ we obtain:

$$\sum_{r \in P \setminus \{1,i\}} (w_{\bar{\alpha}_r} u^{\uparrow}_{\bar{\alpha}_r}(\pi) + w_{\alpha_r} u^{\uparrow}_{\alpha_r}(\pi)) \leq (p-2)(2^{3m} + 2^4 m).$$

Finally, for all remaining agents, we use inequalities $u^{\uparrow}_k(\pi) \leq b_k$ with b_k defined as in (8). This completes the explanation of Equation (14).

Finally, to prove that $\psi_w(u(\pi)) < K$ let us consider the quantity: $K'' = 2^{10m} + 2^{9m} + (s_1 - 1)(2^{7m} + 2^{6m}) + (p-2)(2^{4m} + 2^{3m}) + 2^{3m}$. Using Equation (5) we have:

$$K'' \leq K - 2^{4m}. \tag{15}$$

Moreover, by Equation (14) we have:

$$\psi_w(u(\pi)) - K'' \leq 2^{3m} + (s_1 - 1)2^{2m} + (2^{2m} + 2^m)(6m - 2s_1 + 2p + 2).$$

According to Remark 3, we know that $p \leq 3m/2$ holds; we also assume that $m \geq 5$. Hence we obtain the following inequalities: $2^{3m} + (s_1 - 1)2^{2m} + (2^{2m} + 2^m)(6m - 2s_1 + 2p + 2) < 2^{3m} + m(2^{2m} + 2^{2m+4} + 2^{m+4}) < 2^{3m} + 2^{m-2} \cdot 2^{2m+6} = 2^{3m} + 2^{3m+4} < 2^{3m+5} \leq 2^{4m}$. Therefore, we have:

$$\psi_w(u(\pi)) - K'' < 2^{4m}. \tag{16}$$

Hence, by adding Equations (15) and (16) we obtain $\psi_w(u(\pi)) < K$.

We consider now the case $i = 1$ where the proof follows a similar line with minor variations. In this case, the following inequality is substituted for (14):

$$\begin{aligned}\psi_w(u(\pi)) \leq\ & 2 \cdot 2^{9m} + s_1(2^{7m} + 2^{2m}) + (s_1 - 2)2^{6m} \\ & + (2^{3m} + 2^{4m})(p-1) + (2^{2m} + 2^m)(6m - 2s_1 - 2p + 2).\end{aligned} \tag{17}$$

We have indeed $w_{\bar{\alpha}_1} u^{\uparrow}_{\bar{\alpha}_1}(\pi) + w_{\alpha_1} u^{\uparrow}_{\alpha_1}(\pi) = 2^{3m} \cdot 2^{6m} + 2^0 \cdot 2^{6m+3m} = 2 \cdot 2^{6m+3m}$. Then, using Equation (12) for all $r \in P \setminus \{1\}$ we obtain:

$$\sum_{r \in P \setminus \{1\}} (w_{\bar{\alpha}_r} u^{\uparrow}_{\bar{\alpha}_r}(\pi) + w_{\alpha_r} u^{\uparrow}_{\alpha_r}(\pi)) \leq (p-1)(2^{3m} + 2^{4m}).$$

Finally, for all other terms, we use the same arguments as in the explanation of Equation (14).

Now, to prove that $\psi_w(u(\pi)) < K$ when $i = 1$ let us consider the quantity: $K' = 2^{9m} + (s_1 - 1)(2^{7m} + 2^{6m}) + (p-1)(2^{4m} + 2^{3m})$. Using Equation (5) we have:

$$K' \leq K - 2^{10m}. \tag{18}$$

Moreover, by Equation (17) we have:

$$\psi_w(u(\pi)) - K' \leq 2^{9m} + s_1 2^{2m} + 2^{7m} - 2^{6m} + (2^{2m} + 2^m)(6m - 2s_1 + 2p + 2)$$

As in previous Lemma we have $2p \leq 3m$ and we assume that $m \geq 4$. Hence we obtain the following inequalities: $2^{9m} + s_1 2^{2m} + 2^{7m} - 2^{6m} + (2^{2m} + 2^m)(6m - 2s_1 + 2p + 2) < 2^{9m} + 2^{6m} + m(2^{2m} + 2^{2m+3} + 2^{m+3}) < 2^{9m} + 2^{6m} + m \cdot 2^{2m+5} < 2^{9m} + 2^{6m} + 2^{5m} < 2^{9m} + 2^{7m} < 2^{9m+1} < 2^{10m}$. Therefore, we have:

$$\psi_w(u(\pi)) - K' < 2^{10m}. \tag{19}$$

Hence, by adding Equations (18) and (19) we obtain $\psi_w(u(\pi)) < K$. □

We can now establish the complexity of the FOWAMA problem.

3.3 NP-completeness of FOWAMA

Theorem 1 *The FOWAMA problem is NP-complete.*

Proof First we show that the FOWAMA problem belongs to NP, and for this we have to check that we can compute the OWA value of any assignment in G in polynomial time. For this, we observe that in view of the definition of the utilities and of the weights, all the numbers involved have polynomial size representation (they are all representable with no more than $6pm + 5m + 1$ binary digits). Moreover, since all the weights are powers of 2, each product of a weight and a utility can be obtained as a result of shifting the binary representation of the utility. This shows that (i) the construction of the FOWAMA instance is a polynomial-size many-one reduction having size polynomial in the size of the initial MAX3SAT instance; (ii) computing the OWA value can be done in polynomial time. To show that the problem is NP-complete we use the reduction from MAX3SAT introduced in Subsection 3.1 which is obviously polynomial. Let us consider any instance of MAX3SAT. The question is: is there a truth assignment that simultaneously satisfies at least R clauses in C? To answer this question, we consider the following instance of FOWAMA: is there an assignment π in the graph G such that $\psi_w(u(\pi)) \geq K + 2^{2m}R$, where w is defined by (4) and K is defined by (5)?

First we observe that from any truth assignment of variables $(x_1, \ldots, x_p)$ which satisfies at least R clauses, we can construct a consistent assignment π such that $\psi_w(u(\pi)) \geq K + 2^{2m}R$. This is easily proved by using similar arguments as in the proof of Lemma 1.

Conversely, let us prove that if an assignment π exists such that $\psi_w(u(\pi)) \geq K + 2^{2m}R$ then there exists a truth assignment of variables $(x_1, \ldots, x_p)$ which satisfies at least R clauses. Since the overall value is greater than K, we know that π is consistent by Lemma 4. By Lemma 1 we know that $\psi_w(u(\pi)) = K + 2^{2m}q^\pi$. Since $\psi_w(u(\pi)) \geq K + 2^{2m}R$ we obtain $R \leq q^\pi$. Then, the truth-assignment x^π associated with the consistent assignment π makes at least q^π clauses true. Therefore, at least R clauses are true in this assignment, and this completes the proof of Theorem 1. □

The above result suggests that the existence of polynomial-time algorithms for exactly solving the fair OWA assignment problem is highly unlikely. This therefore makes the investigation of polynomial-time solvable special cases particularly worth considering. The purpose of the next section is precisely to exhibit a wide class of such problem instances.

4 *k*-FOWAMA: Fair OWA assignment with weights featuring at most *k* distinct values

In this section we restrict our attention to the subclass of instances of the FOWAMA problem for which the OWA weight vector w is non-increasing and includes a number of distinct values bounded by a fixed constant k. Let $\mu = (\mu_1, \ldots, \mu_k) \in \mathbb{R}^k$ be the vector of distinct values appearing in the sequence $(w_i)_{i \in N}$. Let $\lambda_0 = 0$ and, for $i \in K = \{1, \ldots, k\}$, let λ_i be the index of the rightmost occurrence of μ_i in this sequence. By definition, the vector $\lambda = (\lambda_1, \ldots, \lambda_k)$ is such that

$\lambda_i < \lambda_{i+1}$, $i = 1, \ldots, k-1$. Moreover, w is such that for any $i \in K$ and any $j \in \{\lambda_{i-1}+1, \ldots, \lambda_i\}$, $w_j = \mu_i$. Observe that it may be the case that $\lambda_k < n$; this means that $w_i = 0$ for all $i \in \{\lambda_k + 1, \ldots, n\}$.

Under these assumptions about the weights, the OWA function ψ_w can be reformulated using μ and λ as follows:

$$\psi_{\mu,\lambda}(u(\pi)) = \sum_{i=1}^{k} \sum_{j=\lambda_{i-1}+1}^{\lambda_i} \mu_i u_j^{\uparrow}(\pi). \tag{20}$$

Now, we propose an algorithm for the k-FOWAMA problem to find an assignment π maximizing $\psi_{\mu,\lambda}(u(\pi))$, where the utility matrix U is supposed to be given.

4.1 A recursive algorithm solving the k-FOWAMA problem

Starting from the initial instance of size n with k distinct weights, the algorithm is recursive and the current step consists in splitting the problem into at most n^2 subproblems of size n, each involving only $k-1$ distinct weights. Then instances of size $k-1$ are recursively split until $k = 1$. The optimum is then recovered from the optimal solutions of subproblems. The number of calls to the main procedure is bounded above by n^{2k} which remains polynomial for fixed k. We first explain the base case $k = 1$ and eventually consider the general case.

The case $k = 1$ corresponds to a situation where the λ_1 first weights of the OWA operator equal μ_1, the others (if $\lambda_1 < n$) being equal to 0. This base case is solved by maximizing $L_{\lambda_1}(u(\pi)) = \sum_{i=1}^{\lambda_1} u_i^{\uparrow}(\pi)$, i.e., the λ_1^{th} component of the Lorenz vector $L(u(\pi))$, for $\pi \in \Pi(N)$. This base problem can be solved in polynomial time, as shown in Gupta et al. [17], using the following procedure: for any threshold value t for utilities, one considers the variant $\hat{u}^t$ of u where all utility values greater than t are set to t. Then if π^t denotes the max-sum assignment (computed e.g. with the Hungarian algorithm) for utility $\hat{u}^t$, then $L_{\lambda_1}(u(\pi^t)) \geq L_{\lambda_1}(u(\pi))$ for any $\pi \in \{\pi' \in \Pi(N) \ : \ u_{\lambda_1}^{\uparrow}(\pi') = t\}$, for more details see Theorem 1 in Gupta et al. [17]. Since $\Pi(N)$ is nothing else but the union of sets $\{\pi' \in \Pi(N) \ : \ u_{\lambda_1}^{\uparrow}(\pi') = t\}$ for all utility values t, the maximization of $L_{\lambda_1}(u(\pi^t))$ over all t yields the optimal assignment π^*. Indeed, for any other assignment π, we have $L_{\lambda_1}(u(\pi)) \leq L_{\lambda_1}(u(\pi^t))$ for $t = u_{\lambda_1}^{\uparrow}$ and therefore, $L_{\lambda_1}(u(\pi)) \leq L_{\lambda_1}(u(\pi^*))$. With our notation, the algorithm is described by Algorithm 1 below.

When $k > 1$ we have to optimize an OWA with k strictly positive distinct values $(\mu_1, \ldots, \mu_k)$ appearing in the sequence $(w_i)_{i \in N}$. We want to reduce the problem to a family of OWA assignment subproblems with $k-1$ distinct values $(\hat{\mu}_1, \ldots, \hat{\mu}_{k-1})$. To this end we consider subproblems where $\hat{\mu}_i = \mu_i / \mu_{k-1}$ for $i = 1, \ldots, k-1$. The subproblems are characterized by different variants of the utility functions defined as follows: for any utility value t, one considers the variant $\hat{u}^t$ of u where all utility values smaller than or equal to t are redefined as $\hat{u}_i^t(o_j) = \mu_{k-1} u_i(o_j) - t(\mu_{k-1} - \mu_k)$ and all utility values greater than t are redefined as $\hat{u}_i^t(o_j) = \mu_k u_i(o_j)$. Denoting π^t the assignment maximizing $\psi_{\hat{\mu},\hat{\lambda}}(\hat{u}^t(\pi))$, the idea behind this transformation is that the optimal assignment with respect to $\psi_{\mu,\lambda}$ is the one that maximizes $\psi_{\mu,\lambda}(\pi^t)$ over t; this argument will be proved in the next

Algorithm 1: LORENZ(u, λ, n)

```
Data: u_1, ..., u_n the individual utility functions
      μ ∈ R^k_+, λ ∈ N^k representing w ∈ R^k_+
Result: π* ∈ Π(N) maximizing L_{λ_1}(u(π))
foreach i ∈ N do
    π*(i) ← i;
end
foreach t such that ∃(i, j) ∈ N^2, u_i(o_j) = t do
    foreach i ∈ N do
        foreach j ∈ N do
            if u_i(o_j) ≥ t then
                û^t_i(o_j) ← t;
            else
                û^t_i(o_j) ← u_i(o_j);
            end
        end
    end
    π^t ← arg max_{π∈Π(N)} Σ_{i=1}^n û^t_i(π);
    if L_{λ_1}(u(π^t)) > L_{λ_1}(u(π*)) then
        π* ← π^t;
    end
end
return π*;
```

section. For the sake of efficiency, we restrict the search to those solutions π^t such that $t \leq v = \max_{\pi\in\Pi(N)}\{u^{\uparrow}_{\lambda_{k-1}}(\pi)\}$ without losing optimality as will be shown later. This value v represents the highest utility value achievable by an assignment for the λ_{k-1}-th most satisfied agent. It can be efficiently computed by applying an algorithm proposed in Gorski et al. [14]. Algorithm 2 given below provides the pseudo-code of our algorithm for OWA optimization.

An illustrative example. Before establishing the correctness of Algorithm 2, we illustrate its behavior on the instance of the FOWAMA problem characterized by matrix U given in Equation (1) and the weighting vector $w = (2, 2, 1, 1, 1)$ to clarify the sequence of recursive calls. Hence we consider a call to SOLVE with the parameters $k = 2$ (number of different weights), $\mu = (\mu_1, \mu_2) = (2, 1)$ (values of weights occurring in w) and $\lambda = (\lambda_1, \lambda_2) = (2, 5)$ (last index values before a change in the weight coefficients). The value of v defined in our algorithm is normally computed with the algorithm presented in Gorski et al. [14]. Here we can guess the result ($v = 11$) since no more than three agents can have a utility strictly higher than 11 and the assignment $\pi'(i) = i$ for $i = 1, \ldots, 5$ is such that $u^{\uparrow}_2(\pi') = 11$. Hence the different values of t considered within the main loop are $\{5, 6, 7, 8, 11\}$. The value of $\hat{\mu}$ and $\hat{\lambda}$ are (1) and (5), respectively (we keep the parenthesis to recall that $\hat{\lambda}$ and $\hat{\mu}$ are vectors). For $t = 11$, we have $s = (\mu_1 - \mu_2)t/\mu_1 = 11/2$ and $\hat{u}^{11}_i(o_j) = \mu_1(u_i(o_j) - s)$ if $u_i(o_j) \leq t$ and $\mu_2 u_i(o_j)$ if $u_i(o_j) > t$. The resulting matrix $\hat{U}^{11}$ is shown below. The call to SOLVE with $k = 1$, $\hat{\mu} = (1)$ and $\hat{\lambda} = (5)$ corresponds to an OWA maximization with weights $(1, 1, 1, 1, 1)$ (i.e., maximizing the sum of the five utilities) which yields π^{11} equal here to the identity assignment. Hence $\pi^* = \pi^{11}$. For $t = 8, 7, 6$ it can be checked that the call to SOLVE with $k = 1$, $\hat{\mu} = (1)$, $\hat{\lambda} = (5)$ and $\hat{u}^t$ as given below also produces the identity assignment and π^* remains unchanged. Finally for $t = 5$,

Algorithm 2: $\textsc{Solve}(u, \mu, \lambda, n, k)$

```
Data: u_1, ..., u_n the individual utility functions
      μ ∈ ℝ^k_+, λ ∈ N^k representing w ∈ ℝ^n_+
Result: π* maximizing ψ_{μ,λ}(u(π)) over π ∈ Π(N)
if k=1 then
    π* ← LORENZ(u, λ, n)
else
    v ← max_{π∈Π(N)} {u↑_{λ_{k−1}}(π)};
    foreach i ∈ {1, ... k − 2} do
        μ̂_i ← μ_i / μ_{k−1};
        λ̂_i ← λ_i;
    end
    μ̂_{k−1} ← 1, λ̂_{k−1} ← λ_k;
    foreach i ∈ N do
        π*(i) ← i;
    end
    foreach t ≤ v such that ∃(i, j) ∈ N², u_i(o_j) = t do
        s ← ((μ_{k−1} − μ_k) / μ_{k−1}) t;
        foreach i ∈ N do
            foreach j ∈ N do
                if u_i(o_j) ≤ t then
                    û^t_i(o_j) ← μ_{k−1}(u_i(o_j) − s);
                else
                    û^t_i(o_j) ← μ_k u_i(o_j);
                end
            end
        end
        π^t ← SOLVE(û^t, μ̂, λ̂, n, k − 1);
        if ψ_{μ,λ}(u(π^t)) > ψ_{μ,λ}(u(π*)) then
            π* ← π^t;
        end
    end
end
return π*;
```

the utility function $\hat{u}^5$ is exactly the same as the original utility function u. So the call to SOLVE provides the assignment π^5 such that $\pi^5(1) = 2$, $\pi^5(2) = 1$ and $\pi^5(i) = i$ for $i = 3, 4, 5$. Since $\psi_{\lambda,\mu}(u(\pi^5)) = 66 < \psi_{\lambda,\mu}(u(\pi^*)) = 71$, the assignment π^* remains unchanged. So at the end, the solution returned by the algorithm is π^* such that $\pi^*(i) = i$ for $i = 1, \ldots, 5$, which indeed corresponds to the ψ_w-optimal solution for $w = (2, 2, 1, 1, 1)$.

$$\hat{U}^{11} = \begin{pmatrix} 12 & 20 & 1 & -1 & 5 \\ -1 & 12 & 1 & 5 & -1 \\ 5 & -1 & 11 & -1 & 1 \\ 1 & 5 & 1 & 11 & -1 \\ -1 & 1 & 5 & 3 & 3 \end{pmatrix} \quad \hat{U}^{8} = \begin{pmatrix} 12 & 20 & 4 & 2 & 8 \\ 2 & 12 & 4 & 8 & 2 \\ 5 & 2 & 11 & 2 & 4 \\ 4 & 8 & 4 & 11 & 2 \\ 2 & 4 & 8 & 6 & 6 \end{pmatrix}$$

$$\hat{U}^{7} = \begin{pmatrix} 12 & 20 & 5 & 3 & 8 \\ 3 & 12 & 5 & 8 & 3 \\ 8 & 3 & 11 & 3 & 5 \\ 5 & 8 & 5 & 11 & 3 \\ 3 & 5 & 8 & 7 & 7 \end{pmatrix} \quad \hat{U}^{6} = \begin{pmatrix} 12 & 20 & 6 & 4 & 8 \\ 4 & 12 & 6 & 8 & 4 \\ 8 & 4 & 11 & 4 & 6 \\ 6 & 8 & 6 & 11 & 4 \\ 4 & 6 & 8 & 7 & 7 \end{pmatrix}$$

4.2 Correctness and complexity of Algorithm 2

Let us first show that Algorithm 2 returns an OWA-optimal assigment. We first establish a preliminary Lemma showing that the transformation of utilities from u to $\hat{u}^t$ preserves the ordering of agents over components, i.e., more formally:

Lemma 5 *For all $\pi \in \Pi(N)$, for all $i, j \in N$, for each utility value t, $u_i(\pi) \geq u_j(\pi) \Rightarrow \hat{u}_i^t(\pi) \geq \hat{u}_j^t(\pi)$.*

Proof Three cases must be distinguished depending on the position of t with respect to $u_i(\pi)$ and $u_j(\pi)$.

Case 1: $t \geq u_i(\pi) \geq u_j(\pi)$. In this case we have: $\hat{u}_i^t(\pi) = \mu_{k-1}(u_i(\pi) - s) \geq \mu_{k-1}(u_j(\pi) - s) = \hat{u}_j^t(\pi)$.

Case 2: $u_i(\pi) > t \geq u_j(\pi)$. In this case we have $\hat{u}_i^t(\pi) = \mu_k u_i(\pi) \geq \mu_k u_j(\pi) = \mu_{k-1} u_j(\pi)(1 - s/t)$ since $\mu_k = \mu_{k-1}(1 - s/t)$. If $s = 0$ we have $\mu_{k-1} u_j(\pi) = \hat{u}_j^t(\pi)$. If $s \neq 0$, $-u_j(\pi)s/t \geq -s$ implies $\mu_{k-1} u_j(\pi)(1 - s/t) \geq \mu_{k-1}(u_j(\pi) - s) = \hat{u}_j^t(\pi)$. Hence $\hat{u}_i^t(\pi) \geq \hat{u}_j^t(\pi)$.

Case 3: $u_i(\pi) \geq u_j(\pi) > t$. In this case we have: $\hat{u}_i^t(\pi) = \mu_k u_i(\pi) \geq \mu_k u_j(\pi) = \hat{u}_j^t(\pi)$.

Remark 4 *For the sake of simplicity, when a particular t is considered, the utility $\hat{u}^t$ is simply denoted $\hat{u}$ hereafter.*

A direct consequence of Lemma 5 is the following:

Proposition 1 *For any utility value t, $\hat{u}^{\uparrow} = (\hat{u}^t)^{\uparrow}$ can be directly derived from $u^{\uparrow}$, for any assignment π, as follows:*

$$\forall i \in N,\ \hat{u}_i^{\uparrow}(\pi) = \begin{cases} \mu_{k-1}(u_i^{\uparrow}(\pi) - s) & \textit{if } u_i^{\uparrow}(\pi) \leq t \\ \mu_k u_i^{\uparrow}(\pi) & \textit{otherwise.} \end{cases} \tag{21}$$

Proof. Lemma 5 shows that the same permutation of agents sorts vectors $u(\pi)$ and $\hat{u}(\pi)$ by increasing order. Hence for all $i \in N$ the transformation from $u_i^{\uparrow}(\pi)$ to $\hat{u}_i^{\uparrow}(\pi)$ given by (21) is identical to the transformation from $u_i(\pi)$ to $\hat{u}_i(\pi)$ performed in Algorithm 2, formally given by:

$$\hat{u}_i(\pi) = \begin{cases} \mu_{k-1}(u_i(\pi) - s) & \text{if } u_i(\pi) \leq t \\ \mu_k u_i(\pi) & \text{otherwise.} \end{cases} \tag{22}$$

□

We establish now two lemmas that will allow us to prove that the output of Algorithm 2 is OWA-optimal with respect to the initial vector of weights (characterized by μ and λ).

Lemma 6 *Let $v = \max_{\pi \in \Pi(N)} \{u_{\lambda_{k-1}}^{\uparrow}(\pi)\}$ as defined in Algorithm 2 then allocation π^v maximizes $\psi_{\mu,\lambda}(u(\pi))$ over the set $\{\pi \in \Pi(N) | u_{\lambda_{k-1}+1}^{\uparrow}(\pi) \geq v\}$.*

Proof Let π be an assignment such that $u_{\lambda_{k-1}+1}^{\uparrow}(\pi) \geq v$. We have to show that: $\psi_{\mu,\lambda}(u(\pi^v)) \geq \psi_{\mu,\lambda}(u(\pi))$. In Algorithm 2, π^v is obtained by a call to

SOLVE$(\hat{u}^v, \hat{\mu}, \hat{\lambda}, n, k-1)$. Hence π^v maximizes $\psi_{\hat{\mu},\hat{\lambda}}(\hat{u}(\pi'))$ for $\pi' \in \Pi(N)$. In particular, for π we have, using (20):

$$\sum_{i=1}^{k-1} \sum_{j=\hat{\lambda}_{i-1}+1}^{\hat{\lambda}_i} \hat{\mu}_i \hat{u}_j^{\uparrow}(\pi^v) \geq \sum_{i=1}^{k-1} \sum_{j=\hat{\lambda}_{i-1}+1}^{\hat{\lambda}_i} \hat{\mu}_i \hat{u}_j^{\uparrow}(\pi). \tag{23}$$

By definition of v, $u_{\lambda_{k-1}}^{\uparrow}(\pi) \leq v$ which implies $u_j^{\uparrow}(\pi) \leq v$ for all $j \in \{1, \ldots, \lambda_{k-1}\}$ since $u_j^{\uparrow}(\pi) \leq u_{\lambda_{k-1}}^{\uparrow}(\pi)$. Similarly we obtain that $u_j^{\uparrow}(\pi^v) \leq v$. So by (21) we have $\hat{u}_j^{\uparrow}(\pi^v) = \mu_{k-1}(u_j^{\uparrow}(\pi^v) - s)$ and $\hat{u}_j^{\uparrow}(\pi) = \mu_{k-1}(u_j^{\uparrow}(\pi) - s)$ for all $j \in \{1, \ldots, \lambda_{k-1}\}$. We also know that $\forall i \in \{1, \ldots, k-1\}$, $\hat{\mu}_i = \frac{\mu_i}{\mu_{k-1}}$. Note also that $\hat{\lambda}_i = \lambda_i$ for all $i \in \{1, \ldots, k-2\}$. Hence we can rewrite the left hand side of (23) as follows:

$$\begin{aligned} L = \sum_{i=1}^{k-1} \sum_{j=\hat{\lambda}_{i-1}+1}^{\hat{\lambda}_i} \hat{\mu}_i \hat{u}_j^{\uparrow}(\pi^v) &= \sum_{i=1}^{k-2} \sum_{j=\hat{\lambda}_{i-1}+1}^{\hat{\lambda}_i} \hat{\mu}_i \hat{u}_j^{\uparrow}(\pi^v) + \sum_{j=\hat{\lambda}_{k-2}+1}^{\hat{\lambda}_{k-1}} \hat{\mu}_{k-1} \hat{u}_j^{\uparrow}(\pi^v) \\ &= \sum_{i=1}^{k-2} \sum_{j=\lambda_{i-1}+1}^{\lambda_i} \mu_i (u_j^{\uparrow}(\pi^v) - s) + \sum_{j=\hat{\lambda}_{k-2}+1}^{\hat{\lambda}_{k-1}} \hat{u}_j^{\uparrow}(\pi^v). \end{aligned} \tag{24}$$

Moreover, since $\hat{\lambda}_{k-1} = \lambda_k > \lambda_{k-1}$ we have:

$$\begin{aligned} \sum_{j=\hat{\lambda}_{k-2}+1}^{\hat{\lambda}_{k-1}} \hat{u}_j^{\uparrow}(\pi^v) = \sum_{j=\lambda_{k-2}+1}^{\lambda_k} \hat{u}_j^{\uparrow}(\pi^v) &= \sum_{j=\lambda_{k-2}+1}^{\lambda_{k-1}} \hat{u}_j^{\uparrow}(\pi^v) + \sum_{j=\lambda_{k-1}+1}^{\lambda_k} \hat{u}_j^{\uparrow}(\pi^v) \\ &= \sum_{j=\lambda_{k-2}+1}^{\lambda_{k-1}} \mu_{k-1}(u_j^{\uparrow}(\pi^v) - s) + \sum_{j=\lambda_{k-1}+1}^{\lambda_k} \hat{u}_j^{\uparrow}(\pi^v). \end{aligned}$$

Hence by substituting the rightmost term of line (24) by the latter line, the left-hand side of Equation (23) becomes:

$$L = \sum_{i=1}^{k-1} \sum_{j=\lambda_{i-1}+1}^{\lambda_i} \mu_i (u_j^{\uparrow}(\pi^v) - s) + \sum_{j=\lambda_{k-1}+1}^{\lambda_k} \hat{u}_j^{\uparrow}(\pi^v).$$

Similarly, we can prove that the right-hand side of Equation (23) is:

$$R = \sum_{i=1}^{k-1} \sum_{j=\lambda_{i-1}+1}^{\lambda_i} \mu_i (u_j^{\uparrow}(\pi) - s) + \sum_{j=\lambda_{k-1}+1}^{\lambda_k} \hat{u}_j^{\uparrow}(\pi).$$

Hence we obtain $L \geq R$ by Equation (23).

Note now that by definition of π we have $u_j^{\uparrow}(\pi) \geq u_{\lambda_{k-1}+1}^{\uparrow}(\pi) \geq v$ for any $j \in \{\lambda_{k-1}+1, \ldots, \lambda_k\}$. So by Equation (21) applied with $t = v$ we have $\hat{u}_j^{\uparrow}(\pi) = \mu_k u_j^{\uparrow}(\pi)$ for any $j \in \{\lambda_{k-1}+1, \ldots, \lambda_k\}$. Therefore, the rightmost term of R can be rewritten using the following equality:

$$\sum_{j=\lambda_{k-1}+1}^{\lambda_k} \hat{u}_j^{\uparrow}(\pi) = \sum_{j=\lambda_{k-1}+1}^{\lambda_k} \mu_k u_j^{\uparrow}(\pi). \tag{25}$$

which gives the following reformulation:

$$R = \psi_{\mu,\lambda}(u(\pi)) - S \quad \text{where} \quad S = \sum_{i=1}^{k-1} \sum_{j=\lambda_{i-1}+1}^{\lambda_i} \mu_i s.$$

Moreover, we can state that for any $j \in \{\lambda_{k-1}+1, \ldots, \lambda_k\}$ we have $\hat{u}_j^{\uparrow}(\pi^v) \leq \mu_k u_j^{\uparrow}(\pi^v)$. To this end, two possible cases must be distinguished depending on the value of $u_j^{\uparrow}(\pi^v)$. If $u_j^{\uparrow}(\pi^v) > v$ then by (21) applied with $t = v$, we have $\hat{u}_j^{\uparrow}(\pi^v) = \mu_k u_j^{\uparrow}(\pi^v)$. On the other hand, if $u_j^{\uparrow}(\pi^v) \leq v$ then by (21) we have $\hat{u}_j^{\uparrow}(\pi^v) = \mu_{k-1}(u_j^{\uparrow}(\pi^v) - s)$. Furthermore, since $s = \frac{\mu_{k-1}-\mu_k}{\mu_{k-1}} t$ and $t = v$ we have:

$$\begin{aligned} &\mu_{k-1}(u_j^{\uparrow}(\pi^v) - s) = \mu_{k-1} u_j^{\uparrow}(\pi^v) - (\mu_{k-1} - \mu_k) v \\ &\leq \mu_{k-1} u_j^{\uparrow}(\pi^v) - (\mu_{k-1} - \mu_k) u_j^{\uparrow}(\pi^v) = \mu_k u_j^{\uparrow}(\pi^v). \end{aligned}$$

where the inequality comes from the fact that $\mu_{k-1} > \mu_k$ and $u_j^{\uparrow}(\pi^v) \leq v$. So it is clear that $\hat{u}_j^{\uparrow}(\pi^v) \leq \mu_k u_j^{\uparrow}(\pi^v)$ for any $j \in \{\lambda_{k-1}+1, \ldots, \lambda_k\}$. Therefore, the rightmost term of L can be lower bounded using the following inequality:

$$\sum_{j=\lambda_{k-1}+1}^{\lambda_k} \mu_k u_j^{\uparrow}(\pi^v) \geq \sum_{j=\lambda_{k-1}+1}^{\lambda_k} \hat{u}_j^{\uparrow}(\pi^v). \tag{26}$$

Hence we obtain $\psi_{\mu,\lambda}(u(\pi)) - S \geq L$ which, in addition to $L \geq R$, completes the proof. □

Lemma 7 *Let $t < v$ such that $t = u_i(o_j)$ for some $i, j \in N$, then allocation π^t as defined in Algorithm 2 maximizes $\psi_{\mu,\lambda}(u(\pi))$ over the set $\{\pi \in \Pi(N) | u^{\uparrow}_{\lambda_{k-1}+1}(\pi) = t\}$.*

Proof Let π be an allocation such that $u^{\uparrow}_{\lambda_{k-1}+1}(\pi) = t$. We have to show that: $\psi_{\mu,\lambda}(u(\pi^t)) \geq \psi_{\mu,\lambda}(u(\pi))$. In Algorithm 2, π^t is obtained by a call to $\textsc{Solve}(\hat{u}^t, \hat{\mu}, \hat{\lambda}, n, k-1)$. Hence π^t maximizes $\psi_{\hat{\mu},\hat{\lambda}}(\hat{u}(\pi'))$ for $\pi' \in \Pi(N)$. In particular, for π we have using (20):

$$\sum_{i=1}^{k-1} \sum_{j=\hat{\lambda}_{i-1}+1}^{\hat{\lambda}_i} \hat{\mu}_i \hat{u}_j^{\uparrow}(\pi^t) \geq \sum_{i=1}^{k-1} \sum_{j=\hat{\lambda}_{i-1}+1}^{\hat{\lambda}_i} \hat{\mu}_i \hat{u}_j^{\uparrow}(\pi). \tag{27}$$

Furthermore, by definition of π, we know that for any $j \in \{1, \ldots, \lambda_{k-1}\}$, we have $u_j^{\uparrow}(\pi) \leq u^{\uparrow}_{\lambda_{k-1}+1}(\pi) = t$. Hence, as in the proof of Lemma 6, the right hand-side of Equation (27) can be rewritten as :

$$R = \sum_{i=1}^{k-1} \sum_{j=\lambda_{i-1}+1}^{\lambda_i} \mu_i (u_j^{\uparrow}(\pi) - s) + \sum_{j=\lambda_{k-1}+1}^{\lambda_k} \hat{u}_j^{\uparrow}(\pi).$$

Moreover, by definition of π, we know that for any $j \in \{\lambda_{k-1}+1, \ldots, \lambda_k\}$, we have $u_j^{\uparrow}(\pi) \geq u^{\uparrow}_{\lambda_{k-1}+1}(\pi) = t$. So by (21) we can state that $\hat{u}_j^{\uparrow}(\pi) = \mu_k u_j^{\uparrow}(\pi)$ for all $j \in \{\lambda_{k-1}+1, \ldots, \lambda_k\}$). Hence we have :

$$\sum_{j=\lambda_{k-1}+1}^{\lambda_k} \hat{u}_j^{\uparrow}(\pi) = \sum_{j=\lambda_{k-1}+1}^{\lambda_k} \mu_k u_j^{\uparrow}(\pi).$$

which give the following reformulation:

$$R = \psi_{\mu,\lambda}(u(\pi)) - S \quad \text{where} \quad S = \sum_{i=1}^{k-1} \sum_{j=\lambda_{i-1}+1}^{\lambda_i} \mu_i s.$$

Besides, we can state that $\hat{u}_j^{\uparrow}(\pi^t) \leq \mu_{k-1}(u_j^{\uparrow}(\pi^t) - s)$ holds for all $j \in \{1, \ldots, \lambda_{k-1}\}$, and that $\hat{u}_{j'}^{\uparrow}(\pi^t) \leq \mu_k u_{j'}^{\uparrow}(\pi^t)$ holds for all $j' \in \{\lambda_{k-1}+1, \ldots, \lambda_k\}$). There are indeed two possible cases depending on the value of $u_j^{\uparrow}(\pi^t)$ (resp. $u_{j'}^{\uparrow}(\pi^t)$). If $u_j^{\uparrow}(\pi^t) \leq t$ (resp. $u_{j'}^{\uparrow}(\pi^t) > t$) then by (21) we have $\hat{u}_j^{\uparrow}(\pi^t) = \mu_{k-1}(u_j^{\uparrow}(\pi^t) - s)$ (resp. $\hat{u}_{j'}^{\uparrow}(\pi^t) = \mu_k u_{j'}^{\uparrow}(\pi^t)$). On the other hand if $u_j^{\uparrow}(\pi^t) > t$ (resp. $u_{j'}^{\uparrow}(\pi^t) \leq t$) then by (21) we have $\hat{u}_j^{\uparrow}(\pi^t) = \mu_k u_j^{\uparrow}(\pi^t)$ (resp. $\hat{u}_{j'}^{\uparrow}(\pi^t) = \mu_{k-1}(u_{j'}^{\uparrow}(\pi^t) - s)$). Furthermore, since $s = \frac{\mu_{k-1}-\mu_k}{\mu_{k-1}} t$ we have : $\mu_k u_j^{\uparrow}(\pi^t) = \mu_k u_j^{\uparrow}(\pi^t) + (\mu_{k-1} - \mu_{k-1})(u_j^{\uparrow}(\pi^t) - s) = \mu_{k-1}(u_j^{\uparrow}(\pi^t) - s) + (\mu_{k-1} - \mu_k)t + (\mu_k - \mu_{k-1})u_j^{\uparrow}(\pi^t) < \mu_{k-1}(u_j^{\uparrow}(\pi^t) - s) + (\mu_{k-1} - \mu_k)u_j^{\uparrow}(\pi^t) + (\mu_k - \mu_{k-1})u_j^{\uparrow}(\pi^t) = \mu_{k-1}(u_j^{\uparrow}(\pi^t) - s)$ where the inequality is due to $t < u_j^{\uparrow}(\pi^t)$ and $\mu_k < \mu_{k-1}$.

For the same reason we can state the following: $\mu_{k-1}(u_{j'}^{\uparrow}(\pi^t) - s) = \mu_{k-1} u_{j'}^{\uparrow}(\pi^t) - (\mu_{k-1} - \mu_k)t \leq \mu_{k-1} u_{j'}^{\uparrow}(\pi^t) - (\mu_{k-1} - \mu_k)u_{j'}^{\uparrow}(\pi^t) = \mu_k u_{j'}^{\uparrow}(\pi^t)$ where the inequality is due to $u_{j'}^{\uparrow}(\pi^t) \leq t$ and $\mu_{k-1} > \mu_k$. So we have shown that $\hat{u}_j^{\uparrow}(\pi^t) \leq \mu_{k-1}(u_j^{\uparrow}(\pi^t) - s)$ holds for all $j \in \{1, \ldots, \lambda_{k-1}\}$ and $\hat{u}_{j'}^{\uparrow}(\pi^t) \leq \mu_k u_{j'}^{\uparrow}(\pi^t)$ for all $j' \in \{\lambda_{k-1}+1, \ldots, \lambda_k\}$. Hence using substitutions and majorations similar to what is done in the proof of Lemma 6 we derive an upper bound on L, the left hand side of (27):

$$\sum_{i=1}^{k-1} \sum_{j=\lambda_{i-1}+1}^{\lambda_i} \mu_i(u_j^{\uparrow}(\pi^t) - s) + \sum_{j=\lambda_{k-1}+1}^{\lambda_k} \mu_k u_j^{\uparrow}(\pi^t) \geq L.$$

Therefore, we have $\psi_{\mu,\lambda}(u(\pi^t)) - S \geq L$ which, in addition to $L \geq R$, completes the proof. □

Now, thanks to the previous two lemmas, we are able to provide the time complexity for Algorithm 2 to solve an OWA optimal assignment provided that OWA weights are non-increasing and include no more than k distinct values.

Theorem 2 *Algorithm 2 returns in time $O(n^{2k+3})$ an assignment maximizing $\psi_{\mu,\lambda}(u(\pi))$ over $\Pi(N)$ where k is the number of distinct weights.*

Proof First we prove that $\pi^* \in \arg\max_{\pi \in \Pi(N)} \psi_{\mu,\lambda}(u(\pi))$. Let $\pi \in \Pi(N)$. If $u_{\lambda_{k-1}+1}^{\uparrow}(\pi) \geq v$ then, by Lemma 6 we know that $\psi_{\mu,\lambda}(u(\pi^v)) \geq \psi_{\mu,\lambda}(u(\pi))$, where π^v is the assignment found at step $t = v$. If $u_{\lambda_{k-1}+1}^{\uparrow}(\pi) < v$ then by Lemma 7 we know that $\psi_{\mu,\lambda}(u(\pi^t)) \geq \psi_{\mu,\lambda}(u(\pi))$, where π^t is the assignment found at step $t = u_{\lambda_{k-1}+1}^{\uparrow}(\pi)$. Finally, we know that the assignment π^* returned by Algorithm 2 is such that $\psi_{\mu,\lambda}(u(\pi^*)) \geq \psi_{\mu,\lambda}(u(\pi^t))$ for any assignment π^t founded during the algorithm. Hence we clearly have $\psi_{\mu,\lambda}(u(\pi^*)) \geq \psi_{\mu,\lambda}(u(\pi))$ for any $\pi \in \Pi(N)$.

Let us show now, by induction on k that Algorithm 2 runs in $O(n^{2k+3})$. For $k = 1$, Algorithm 1 solves the problem in $O(n^5)$ according [17], assuming that standard assignment subproblems are solved in $O(n^3)$ e.g. using the Edmonds and Karp algorithm [9]. So the hypothesis holds for $k = 1$. Now, assume that the complexity of finding an assignment which maximizes an OWA with $k-1$ different weights is $O(n^{2(k-1)+3})$. In Algorithm 2, the value $v = \max_{\pi \in \Pi(N)} \{u^{\uparrow}_{\lambda_{k-1}}(\pi)\}$ can be computed in $O(n^3 \log(n))$ using the algorithm proposed in Gorski et al. [14]. In Algorithm 2 the first loop is performed in linear time. For the second loop, the number of possible values for t is bounded above by n^2. The number of steps for the inner loop is bounded above by n^2. Then comes a call to $\textsc{Solve}(\hat{u}^t, \hat{\mu}, \hat{\lambda}, n, k-1)$ that runs in $O(n^{2(k-1)+3})$ by induction hypothesis. Overall, the complexity of the main loop is $O(n^{2k+3})$. This is also the complexity of the whole algorithm. □

Therefore, the complexity of Algorithm 2 is exponential in k, but the algorithm becomes polynomial for k bounded by a fixed constant. This algorithm shows that FOWAMA belongs to the complexity class XP. However, membership of FOWAMA in FPT (i.e., deciding whether this problem can be solved in time $n^c 2^{O(k)}$, where c is a constant value) remains an open problem.

5 Conclusion

In the first part of the paper, we have proved NP-completeness of FOWAMA, the decision problem associated with the search for an OWA-optimal assignment. In spite of the existence of many known polynomial cases, this issue has been an open problem until now. It is worth noting that this result also implies NP-hardness of the optimal assignment problem w.r.t. a concave Choquet integral (see e.g. [22]) since it includes FOWAMA as a special case.

In the second part of the paper, we have proposed a polynomial time algorithm to solve the fair OWA multiagent assignment problem under some restriction on the weight vector. This result shows that FOWAMA belongs to XP but membership in FPT remains an open question. Besides this, another question deserving investigation would be to study polynomial cases for the determination of an optimal assignment with respect to other aggregation functions, more general than OWA, and used to express fairness, e.g., weighted OWA [34] and Choquet integrals [30].

Acknowledgements

The referees are gratefully acknowledged for their constructive comments and suggestions which resulted in an improved presentation of the paper.

References

1. Aziz, H., Gaspers, S., Mackenzie, S., Walsh, T.: Fair assignment of indivisible objects under ordinal preferences. In: Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems, pp. 1305–1312 (2014)
2. Boland, N., Domínguez-Marín, P., Nickel, S., Puerto, J.: Exact procedures for solving the discrete ordered median problem. Computers & OR **33**(11), 3270–3300 (2006)

3. Bouveret, S., Lang, J.: Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. Journal of Artificial Intelligence Research **32**, 525–564 (2008)
4. Bouveret, S., Lemaître, M., Fargier, H., Lang, J.: Allocation of indivisible goods: a general model and some complexity results. In: Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1309–1310 (2005)
5. Chassein, A., Goerigk, M.: Alternative formulations for the ordered weighted averaging objective. Information Processing Letters **115**(6), 604–608 (2015)
6. Chevaleyre, Y., Dunne, P., Endriss, U., Lang, J., Lemaître, M., Maudet, N., Padget, J., Phelps, S., Rodríguez-Aguilar, J., Sousa, P.: Issues in multiagent resource allocation. Informatica (Slovenia) **30**(1), 3–31 (2006)
7. Chong, K.: An induction theorem for rearrangements. Candadian Journal of Mathematics **28**, 154–160 (1976)
8. Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, pp. 151–158. ACM (1971)
9. Edmonds, J., Karp, R.: Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM **19**(2), 248–264 (1972)
10. Garey, M., Johnson, D.: Computers and intractability. W.H. Freeman and company (1979)
11. Garfinkel, R.: An improved algorithm for the bottleneck assignment problem. Operations Research **19**(7), 1747–1751 (1971)
12. Garg, N., Kavitha, T., Kumar, A., Mehlhorn, K., Mestre, J.: Assigning papers to referees. Algorithmica **58**, 119–136 (2010)
13. Golden, B., Perny, P.: Infinite order Lorenz dominance for fair multiagent optimization. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, pp. 383–390 (2010)
14. Gorski, J., Ruzika, S.: On k-max-optimization. Operations Research Letters **37**(1), 23–26 (2009)
15. Gourvès, L., Monnot, J., Tlilane, L.: A matroid approach to the worst case allocation of indivisible goods. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 136–142 (2013)
16. Gupta, S., Punnen, A.: Minimum deviation problems. Operations Research Letters **7**(4), 201–204 (1988)
17. Gupta, S., Punnen, A.: k-sum optimization problems. Operations Research Letters **9**(2), 121–126 (1990)
18. Hardy, G., Littlewood, J.E., Pólya, G.: Inequalities. Cambridge University Press (1934)
19. Heinen, T., Nguyen, N.T., Rothe, J.: Fairness and rank-weighted utilitarianism in resource allocation. In: Proceedings of the 4th International Conference on Algorithmic Decision Theory, pp. 521–536 (2015)
20. Kalcsics, J., Nickel, S., Puerto, P., A., T.: Algorithmic results for ordered median problems. Operations Research Letters **30**(3), 149–158 (2002)
21. Kuhn, H.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
22. Lesca, J., Minoux, M., Perny, P.: Compact versus noncompact LP formulations for minimizing convex Choquet integrals. Discrete Applied Mathematics **161**(1-2), 184–199 (2013)
23. Lesca, J., Perny, P.: LP Solvable Models for Multiagent Fair Allocation problems. In: Proceedings of the 19th European Conference on Artificial Intelligence, pp. 387–392 (2010)
24. Marshall, W., Olkin, I.: Inequalities: Theory of Majorization and its Applications. Academic Press, London (1979)
25. Minoux, M.: Solving combinatorial problems with combined min-max-min-sum objective and applications. Mathematical Programming **45**(1-3), 361–372 (1989)
26. Moulin, H.: Axioms of cooperative decision making. Monograph of the Econometric Society. Cambridge University Press (1988)
27. Ogryczak, W.: Multicriteria models for fair resource allocation. Control and Cybernetics **36**, 303–332 (2007)
28. Ogryczak, W., Luss, H., Pióro, M., Nace, D., Tomaszewski, A.: Fair optimization and networks: A survey. Journal of Applied Mathematics (2014)
29. Ogryczak, W., Sliwinski, T.: On solving linear programs with the ordered weighted averaging objective. European Journal of Operational Research **148**(1), 80–91 (2003)
30. Schmeidler, D.: Integral representation without additivity. Proceedings of the American Mathematical Society **97**(2), 255–261 (1986)
31. Sen, A.: On economic inequality. Oxford University Press (1973)

32. Shorrocks, A.: Ranking income distributions. Economica **50**, 3–17 (1983)
33. Sokkalingam, P., Aneja, Y.: Lexicographic bottleneck combinatorial problems. Operations Research Letters **23**(12), 27–33 (1998)
34. Torra, V.: The weighted owa operator. International Journal of Intelligent Systems **12**, 153–166 (1997)
35. Weymark, J.: Generalized Gini inequality indices. Mathematical Social Sciences **1**, 409–430 (1981)
36. Yager, R.: On ordered weighted averaging aggregation operators in multicriteria decision making. In: IEEE Transactions on Systems, Man and Cybernetics, vol. 18, pp. 183–190 (1988)