Multi-Value Alignment in Normative Multi-Agent System: An Evolutionary Optimisation Approach

Maha Riad[a;*], **Vinicius de Carvalho**[b] **and Fatemeh Golpayegani**[a]

[a]School of Computer Science, University College Dublin (UCD)
[b]Escola Politécnica (EP), University of Sao Paulo (USP)
ORCiD ID: Maha Riad https://orcid.org/0000-0003-1128-2579, Vinicius de Carvalho https://orcid.org/0000-0002-4623-7244, Fatemeh Golpayegani https://orcid.org/0000-0002-3712-6550

Abstract. Value-alignment in normative multi-agent systems is used to promote a certain value and to ensure the consistent behaviour of agents in autonomous intelligent systems with human values. However, the current literature is limited to the incorporation of effective norms for single-value alignment with no consideration of agents' heterogeneity and the requirement of simultaneous promotion and alignment of multiple values. This research proposes a multi-value promotion model that uses multi-objective evolutionary algorithms and decentralised reasoning to produce the optimum parametric set of norms that is aligned with multiple simultaneous values of heterogeneous agents and the system.To understand various aspects of this complex problem, several evolutionary algorithms were used to find a set of optimised norm parameters considering two toy tax scenarios with two and five values are considered. The results are analysed from different perspectives to show the impact of a selected evolutionary algorithm on the solution, and the importance of understanding the relation between values when prioritising them.

1 Introduction

Normative multi-agent systems (NorMAS) have been used effectively to coordinate the behaviour of agents in multi-agent systems (MAS) that model complex applications such as intelligent transport systems [16]. The norms in NorMAS are regulative norms defined by a social group to regulate behaviours [22]. For example, in a traffic system, the norm is to give priority to emergency vehicles. Also, it is the norm that passengers leave front seats in buses for senior people.These examples represent guidelines that might be recommended in some societies, obligated, or prohibited [22], so when the agents are aware of the norms of the environment they are operating in, they can synchronise their behaviour with other agents, facilitate group decision making and collaborate. However, it is essential to promote human values in MAS as well to reflect real applications.

In the context of this research, the term 'value' refers to motivational values, which represent standards that serve desirable objectives [24]. In other words, a 'value' will represent a preferred state[24], such as equality, health, fairness, etc. [2]. For example (to differentiate between a 'norm' and a 'value'), as a norm, companies give their employees maternity leaves if they have a newborn baby. However, if the values of one of the companies support equality between men and women, both can have equal maternity leaves [24].

The concept of value-alignment was introduced in [15],[23] and [24], to reflect the alignment of norms and values. Researchers used several techniques to address this challenge, they included: reasoning strategies [2], learning methodologies [19], utility-based approaches[10], and genetic algorithms [15]. However, the proposed solutions neglect one or more of the following points. First, they match the norms with only one value or with the preferred sub-set of values, while in the real world, all the values need to be aligned with the norms. Second, these models might not consider heterogeneous MAS, in which different groups of agents support different values, especially when these values are incompatible. For example, supporting both fairness and equality may be conflicting, as ensuring fairness does not necessarily support equality. Third, some works directly derive norms from the values of the system. However, in many systems norms and values may be incompatible and they should be considered independently. For instance, a community can have a value of supporting equality, at the same time of having a norm of giving priority to senior people in queues, or exempting them from paying taxes. In this paper, we address these limitations by proposing **N**orms **O**ptimisation and **V**alue **A**lignment Model (NOVA), that has three main goals:

1. Choose the *best set of norms* in NorMAS with heterogeneous group of agents.
2. *Optimise multiple values* in NorMAS, these values can be:
   - compatible and *incompatible* values.
   - defined by *heterogeneous* groups of agents.
3. Align independent sets of norms and values in NorMAS.

To reach these goals, we formalised the problem as a multi-objective optimisation problem (MOP), in which we represented the values by objectives that need optimisation, and modelled the norms as the decision variables. This allowed us to get the *best set of norms* (the decision variables) when the values (the objectives) are optimised, and so, aligned the norms and the values. Moreover, solving it as a multi-objective optimisation problem, (i) allowed the system to facilitate *Optimising multiple values* defined by *heterogeneous* groups of agents, and (ii) allowed multiple compatible and *incompatible* values (objectives) *optimisation*.

We proposed to solve this problem using multi-objective evolutionary algorithms (MOEAs) as they have been successfully applied to solve MOPs [3] in several domains including logistics, ride-sharing [6], environmental/economic dispatch (EED) problems [21],

[*] Corresponding Author. Email:maha.riad@ucdconnect.ie.

feature selection for machine learning problems [5], and by optimising antibiotic treatments [20]. We applied several MOEAs (NSGA-II, MOEA/DD, SPEA2, and MOMBI2) on different evaluation scenarios to analyse the performance of each of the MOEAs.

Also, as the MOEAs produce sets of non-dominant optimum solutions, to choose a final solution we extended the agents' logic with a reasoner that allowed them to vote for their preferred solution.

Accordingly, our proposed model NOVA, is a multi-value promotion model that uses multi-objective evolutionary algorithms to produce an optimum parametric set of norms that is aligned with the values of heterogeneous agents' groups. We evaluated NOVA using different scenarios that measure the effect of using different combinations of values. Our contribution is three-fold:

- Multiple values alignment: we show the capability of choosing the optimum values for a parametric norms set while aligning it with a set of multiple optimised values.
- Incompatible and compatible values alignment: we model the problem as a multiple-objective problem to enable the simultaneous optimisation of all values regardless of their compatibility.
- Heterogeneous agents groups' values alignment: we align values from different heterogeneous groups of agents while considering shared system values.

2 Problem Formulation

Let us consider a heterogeneous normative multi-value multi-agent system that is composed of a finite set of regular agents as $Ag = \{ag_1, ag_2, ..., ag_n\}$. Each agent ag_i has a set of values V_{ag_i}, a set of properties Pr_{ag_i}, a set of actions A_{ag_i}, and a set of adopted norms N_{ag_i}. There is one regulative agent r that is responsible to synthesise the norms set N, in which $N_{ag_i} \subseteq N$. The norms are parametric norms, i.e. each norm n_j has a set of parameters P_{n_j} that can contain unbounded or constrained elements with discrete or continuous domains. The regulative agent r has a set of values V_r as well. In each step (iteration) t, each regular agent ag_i performs actions from A_{ag_i} and applies its set of adopted norms N_{ag_i}. The regulative agent also applies actions chosen from its set of actions A_r. Corresponding to the agents' new situations, a global state s_t is captured by r. In such a system, r's main challenges are: to synthesise the optimum set of norms that ensures the alignment of its own values V_r and each of the regular agent's values V_{ag_i} (which is shared between a subset of agents), and to optimise the synthesised set of norms even in case of incompatible values.

2.1 *Defining the Problem as a Multi-Objective Optimisation Problem (MOP)*

As the main aim is to find the best set of parametric norms when the agents' and system's values are satisfied (optimised), we consider the problem as a multi-objective optimisation problem (MOP).

Multi-objective optimisation requires finding solutions which simultaneously consider two or more conflicting objectives to be minimised or maximised [17]. Thus, the optimisation process aims to find a set of solutions that reflects a trade-off between the objectives. MOPs are formulated using: objective functions, constraints, decision variables and their bounds [17].

Respectively, in NOVA, we formulate the problem identified in Section 2 as a multi-objective optimisation problem. We define the agents' and system's values as the objective functions to be optimised, and the norms as the decision variables.

3 Norms Optimisation and Values Alignment Model (NOVA)

NOVA is a model for norms optimisation and values alignment, its main responsibilities are to: (i) optimise the values (objectives), (ii) choose the best set of norms, (iii) reason non-dominant solutions, and (iv) produce one final optimum solution for aligning multiple norms and objectives. As seen in Figure 1, NOVA operates using a main regulative agent r and regular agents Ag. The regulative agent is responsible for: initialising the environment parameters and norms, collecting values from regular agents, and doing the optimisation process using the *Optimiser* component. After the *Optimiser* produces the set of non-dominant solutions, the *Main Reasoner* in r is triggered to start the reasoning process with the regular agents Ag.

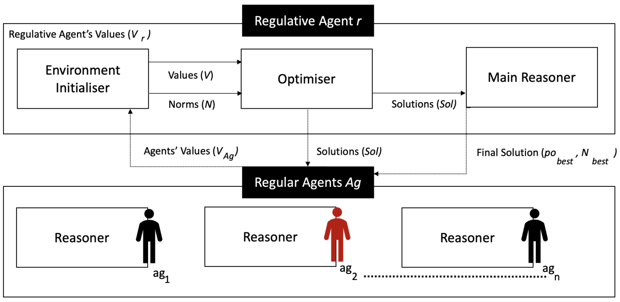

Figure 1. NOVA Conceptual Model

In the next subsections (3.1 and 3.2), we illustrate in more details the two main processes carried by NOVA.

3.1 *Optimisation Process*

The optimisation process is performed by the *Optimiser* in the regulative agent r after the *Environment Initialiser* (i) defines the set of parametric norms and their values bounds (ii) collects values of all agents V_{ag} and integrates them with its values V_r in a single set of values V. (iii) maps the norms as decision variables and the values as the objectives and send them to the *Optimiser*.

Accordingly, NOVA synthesises the best set of norms (decision variables) based on the agents' values (objective functions) optimisation, using the following off-line approach:

Environment Initialisation: initially, the parametric norms set N is initialised with random values between the norms' specified boundaries by the regulative agent r (as seen in Fig. 2). Also, the primary values of the Properties Pr_{ag_i} for each of the agents are defined. The regulative agent r sends the initial set of norms N to the agents. The values of the set of agents Ag in the system and the regulative agent r are consolidated by r in one set of values V. These initial values are used to calculate the global state s_0. Then, the processed N, V_r, s_0, and $MOEA$ (which is the type of the multi-objective evolutionary algorithm that will be used for the optimisation, as illustrated in the next paragraph) are used as input parameters to start the *NOVA optimisation strategy* used in Algorithm 1.

MOEA usage in NOVA. NOVA uses different Multi-Objective Evolutionary Algorithms (MOEAs) to solve the multi-objective optimisation problem, and produce the Pareto Front set of solutions (set of non-dominant solutions). MOEAs are heuristic techniques that provide a flexible representation of the solutions and do not impose continuity conditions on the functions to be optimised. Moreover, MOEAs are extensions of Evolutionary Algorithms (EAs) for

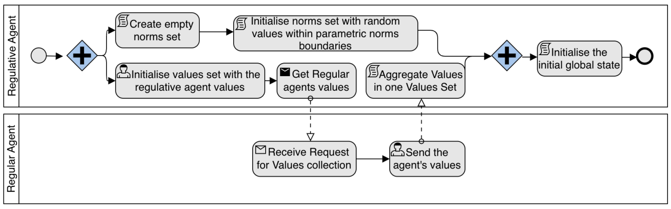
Regulative Agent
Create empty norms set
Initialise norms set with random values within parametric norms boundaries
Initialise the initial global state
Initialise values set with the regulative agent values
Get Regular agents values
Aggregate Values in one Values Set
Regular Agent
Receive Request for Values collection
Send the agent's values

Figure 2. System Initialisation Process Instantiated by the Regulative Entity

multi-objective problems that usually apply the concepts of Pareto dominance [1]. In Pareto dominance, a certain solution sl_a in the decision space of a MOP is superior to another solution sl_b if and only if $f(sl_a)$ is at least as good as $f(sl_b)$ in terms of all the objectives and strictly better than $f(sl_b)$ in terms of at least one single objective. Solution sl_a is also said to strictly dominate solution sl_b [1]. In Nova *Optimiser*, we use four MOEA algorithms: **NSGA-II** [8], **SPEA2** [26] and **MOMBI2**[11] that differ from each other mainly in the way that solutions are ranked at every iteration [25], and **MOEA/DD** [14] which is different in its decomposition technique .

NOVA Optimisation Strategy: as NOVA optimiser is built on a genetic (EA) strategy, it takes the following main steps in each iteration t, see Algorithm 1:

1. Each of the agents in Ag and the regulative agent r carry out their actions while applying the relative norms to these actions. These actions produce a new global state s_t. [Lines 3-6]
2. The regulative agent r performs its actions A_r on s_t considering the current norms N. [Line 7]
3. The regulative agent r uses the new global state to perform the optimisation process using a multi-objective optimiser $MOEA$ and produces the new set of norms N based on the optimised set of values V. [Line 8]
4. The new set of norms N is communicated to all the agents in Ag. [Line 9]

Algorithm 1 NOVA Optimisation Strategy

```
 1: Input: $N$, $V$, $s_0$, $MOEA$
 2: for each t do
 3:     for each $ag_i \in Ag$ do
 4:         $s_t \Leftarrow ag_i.act(N_{ag_i}, A_{ag_i}, s_t)$
 5:     end for
 6:     $s_t \Leftarrow r.act(N, A_r, s_t)$
 7:     $N \Leftarrow r.optimise(s_t, V, N, MOEA)$
 8:     $r.inform(Ag, N)$
 9: end for
10: $N^* \Leftarrow N$
```

3.2 *Reasoning Process*

The multi-objective optimisation process produces the Pareto Front set of solutions PF_{known}, and then, it sends each solution with its corresponding norms set to the *Main Reasoner* (see Figure 1) as Sol, where $sol_j = \{pf_j, N_{ag_i}^{pf_j}\}$. Afterwards, a decentralised reasoning process takes place to produce one final optimum solution sol_{best}.

As indicated in Algorithm 2 and Figure 3, the reasoning process starts by running the $mainReasoner()$ after receiving the Pareto Front set and its corresponding parametric norms, and formulating Sol. First, in line 3, the reasoner creates an empty list to store in it the votes that will be collected from the regular agents. Each of the reasoning (regular) agents is asked to vote in line 4 by calling the $getVote$ method. The $getVote$ method takes as parameters the Pareto Front set of optimum solutions PF_{Known}, and the $N_{ag_i}^{PF_{Known}}$ parametric norm values that correspond to these solutions and belong to this agent's group. In line 10, the preferred decision variable (i.e. norm to be prioritised) is stored in $prefVar$ variable. Depending on the $prefVar$, the norms set that prompts this $prefVar$ the most is stored in $n_{sol_{best}}$. Subsequently, the solution that couples this norms set is saved as the chosen solution pf_{best}. Then, this solution is added to the $votes[]$ at line 5. After calculating the solution with the maximum number of votes, the main reasoner states the final chosen solution in line 7.

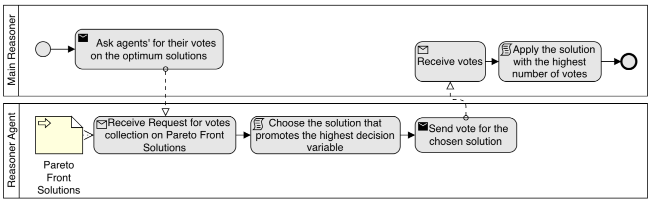
Main Reasoner
Ask agents' for their votes on the optimum solutions
Receive votes
Apply the solution with the highest number of votes
Reasoner Agent
Pareto Front Solutions
Receive Request for votes collection on Pareto Front Solutions
Choose the solution that promotes the highest decision variable
Send vote for the chosen solution

Figure 3. Reasoning Process

Algorithm 2 Reasoning Algorithms

```
 1: Input:, $Sol$
 2: function $mainReasoner()$: $sol_{best}$
 3:     $votes[] \Leftarrow null$
 4:     for each $ag_i \in Ag$ do
 5:         $votes.add(ag_i.getVote(Sol))$
 6:     end for
 7:     $sol_{best} \Leftarrow maxVotesSol(votes)$
 8: end function
 9: function $getVote(PF_{Known}, N_{ag_i}^{PF_{Known}})$: $pf_{best}$
10:     $prefVar \Leftarrow ag_i.getPriorityDecisionVar()$
11:     $n_{sol_{best}} \Leftarrow getMax(prefVar, N_{sol})$
12:     $pf_{best} \Leftarrow getObjectiveSol(n_{sol_{best}})$
13: end function
```

The Voting Process: when the regular agents Ag are reasoning the best solution, they calculate the fitness fit of each solution by calculating equation 1. Wg is the set of weights defined for each of the agent's norms, which is created randomly, and $\sum_i^N Wg_i = 1$. The weights are defined based on the preferred decision variable (norms), by assigning a higher value (such as 0.8) to the preferred variable and splitting the remaining weight (0.2) among the other variables.

$$fit(PF, Wg) = select \max \forall s \in PF \sum_{i}^{qtdVars} Wg_i * Var_i^s \qquad (1)$$

Finally, as it is expected that different agents choose different solutions, the most voted solution is elected and returned as the final solution.

$$vote(Ag) = \sum_{a \in Agents} 1 \text{ if } fit(PF, Ag.Wg) > fit(PF, a.Wg), \quad ag \neq a \qquad (2)$$

$$\max \forall Ag \in Agents, vote(Ag) \qquad (3)$$

4 Tax System Scenario

For further illustration of NOVA and for evaluation we will use an adapted tax system toy scenario introduced in [15]. In this scenario, the regular agents set Ag will represent the set of citizens and the regulating agent r will represent the government. The government collects taxes from the citizens according to their wealth group. There are five wealth groups, the 1^{st} group represents the poorest group while the 5^{th} group represents the richest group. A percentage of the citizens do not pay taxes and will be considered as evaders. However, if they were caught by the government they will be punished and will pay the evaded payment in addition to extra fines. In case they do not have sufficient funds only the available money is collected to avoid getting the citizen into debt. After the taxes and fines are collected a 5% will be considered as a fixed interest rate that is added to the total collected amount. Then, the total collected money *cr* will be redistributed back to the citizens depending on their wealth group. Initially (i.e. before simulation), The wealth of each citizen is randomly assigned after being initialized using a random uniform distribution U(0,100). Then, agents are allocated to their corresponding wealth group, with a constraint that the wealth groups have an equal number of citizens. The main characteristics of the system are as follows. First, each of the citizens has four main properties in their properties set, which describes its current state. The properties are:

- Wealth (w_i): it has a numerical value that represents the amount of money citizen i currently has.
- Wealth group (g_k): it represents the wealth group the citizen belongs to according to its wealth w_i.
- Evader flag (e): it reflects whether this citizen is an evader and will not pay taxes or not.
- Primary Wealth (pw_i): it has a numerical value that represents the wealth of the citizen i at the beginning of a time-step before taking any action and before its state changes.

Second, each citizen has a set of values V_{ag_i}, for simplicity in this example, citizens in the same wealth group have the same fixed set of values. In other words, each wealth group has a set of values V_{g_i}, this could represent the community values. Only it is assumed that wealth group g_2 does not have a value to simulate citizens with no particular values, to see how they are affected by the values encouraged by others. Third, the government has its own set of values as well V_r and a set of parametric norms, which has initial randomly defined values. The norms are defined in the same manner they are stated in [15]:

- **n1** defines the tax rate $collect_j$ each wealth group is expected to pay at each time-step. The parametric set of the norm is defined as P_{n_1} = $\{collect_j\}_{j=1,...5}$. The tax rate values are restricted between 0 and 1.
- **n2** defines the fractional percentage $redistribute_j$ each wealth group will take back from the redistribution amount at the end of each time-step. The parametric set of the norm is defined as P_{n_2} = $\{redistribute_j\}_{j=1,...5}$, the values are between 0 and 1 and the sum of the fractions is constrained to be equal to 1.
- **n3** defines the catch rate of evaders. This single parameter is defined as P_{n_3} = $\{catch\}$. Its value is constrained to be between 0 and 1/2 to reflect the difficulty of law-enforcement tasks.
- **n4** defines the extra fine defined as punishment when an evader is caught. This single parameter is defined as P_{n_4} = $\{fine\}$. However, the total amount to be paid by a caught evader, which is equal to the fine plus the taxes amount, can not exceed the total wealth of the evader.

The main challenge of this system is represented in the government's responsibility to optimise the parameters' sets P_{n_i} of the previously defined four norms belonging to $N = \{n_1, n_2, n_3, n_4\}$, while aligning them with the values of the government, as well as the regular citizen's values. The values are defined as follows.

- **Value 1 (Obj1)**: the value of the government is *Equality*, as it aims to treat all the citizens equally without being biased to any group. *Equality* is calculated using equation 4 introduced in [15]. $GI(s)$ represents the Gini Index of the global state s. The Gini Index [9] is an indicator of inequality, where w_k is the wealth of agent ag_k and $\overline{w}$ is the average wealth of all agents at state s.

$$Equality = 1 - 2.GI(s),\ with\ GI(s) = \frac{\Sigma_{i,j\in Ag} \mid w_i - w_j \mid}{2. \mid Ag \mid^2 .\overline{w}} \tag{4}$$

- **Value 2 (Obj2)**: the value of citizens in wealth group g_3 is *Fairness*. The main aim of this value is to have the highest number of evaders in the wealth group g_1 (the poorest group). To promote the estimated probability P of evaders in g_1 at state s, and to increase fairness, equation 5 is used as suggested in [15].

$$Fairness = 2.P[g_i(s) = 1 \mid evader_i] - 1 \tag{5}$$

- **Value 3 (Obj3)**: the value of citizens in wealth group g_5 is to maximise their *Wealth*. The main aim of this value is to have the maximum wealth portion from the total wealth. It represents the new wealth of the citizens after an iteration takes place. Equation 6 is used for calculating the new wealth.

$$Wealth = \frac{\Sigma_{i\in g_5} w_i}{\Sigma_{j\in Ag} w_j} \tag{6}$$

- **Value 4 (Obj4)**: the value of citizens in wealth group g_4 is to maximise the *Gained Amount*. This value aims to have the maximum gain portion from the common amount available for redistribution cr. The gained value is the difference between the citizen's new wealth w_k and the old wealth pw_k (Check the numerator in equation 7).

$$Gained\ Amount = \frac{\Sigma_{i\in g_4} w_i - pw_i}{cr} \tag{7}$$

- **Value 5 (Obj5)**: the value of citizens in wealth group g_1 is related to the *Collect Portion*. This value aims to have the minimum portion from the collect rate out of 1 (a total portion of collect rates). To inverse this to a maximisation objective we have: 8.

$$Collect\ Portion = 1 - Collect_{g_1} \tag{8}$$

The best alignment between the synthesised set of parametric norms and the values is achieved by maximising these 5 values (objectives).

4.1 Applying the Optimisation Process of NOVA

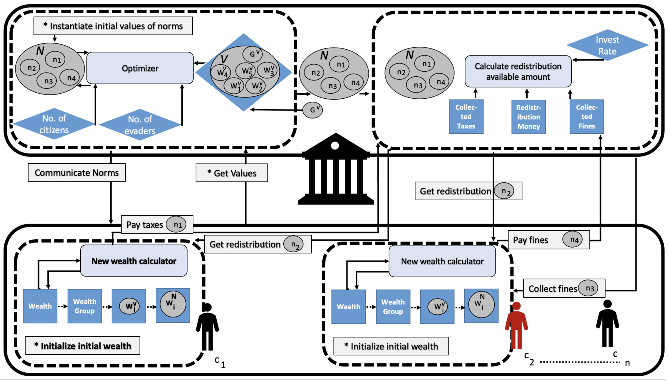

Figure 4. Optimisation Process of NOVA in the TAX System Scenario

In the tax system scenario illustrated in section 4, NOVA's goal is to find the values of the parametric norms n1, n2, n3 and n4 while optimising the values in V: equality, fairness, wealth, gained amount and collect portion. As it is seen in NOVA's conceptual model in Fig. 4, the system is divided into two main divisions, the divisions of the government and citizens. Evaders are represented in red as they have a different set of norms and actions than normal citizens. In this model, first, NOVA randomly initializes the norms and the wealth of the citizens, and consequently, they are assigned to their corresponding wealth groups. Second, the norms set N are communicated by the government (the regulative agent) to the citizens. Third, the citizens start applying the different actions and their corresponding norms. So, normal citizens will start paying taxes according to the rate of their wealth group defined by $n1$. Then the government will start catching the evaders according to the catch value defined in $n3$. The caught evaders will pay their taxes plus the fines determined using $n4$. Afterwards, the government calculates the total amount of money available for redistribution. Subsequently, each citizen receives their portion from redistribution according to the redistribution rate defined by $n2$. Then, the citizens calculate their new wealth and move to their new wealth groups. Forth, based on s_t, the government uses the optimiser to decide the new values of norms by optimising the five values in V. This cycle is repeated until NOVA reaches a stopping condition that represents a satisfying level of the optimisation of the values in V.

4.2 Applying the Reasoning Process of NOVA

In the taxation scenario, the government will carry out the tasks of the *Main Reasoner*. Accordingly, after receiving the solutions set Sol, it will ask the citizens to vote for the best solution Sol_{best}. Each citizen (i.e. each $ag \in Ag$) will randomly choose its preferred decision variable $prefVar$ to prioritise, and will choose the solution that gives the highest value for $prefVar$. If the $prefVar$ supports $n1$ or $n2$ (a parametric set of norms), the citizens will check the highest solution of the norms values that are in P_{n1} or P_{n2} that belongs to their wealth group. For example if an agent ag_1 has its $prefVar$ as $n2$ and belongs to wealth group g_1, it will choose from the solution that has the best value for $P_{n2} = \{redistribute_{j=1}\}$. After all the citizens vote, the solution with the maximum number of votes is set as Sol_{best}.

5 Experimental Evaluation

We evaluated four algorithms NSGA-II, MOEA/DD, SPEA2, and MOMBI2 on solving both two and five objectives problems. Both problems are based on the tax scenario defined in section 4. The two objectives problem includes value 1 (Equality) and value 2 (Fairness), and the five objectives problem includes all the values. Further, we compared the results of the two-objectives scenario with a state-of-the-art work [15], as they tackle the value-alignment problem using genetic algorithm, however they handle only one value per run (i.e. do not support multi-objectives).

Experimental Settings: We used 200 agents to represent the citizens, and a randomly chosen number of evaders in each iteration. The number of segments that represents the wealth groups was set as 5. The investment rate was 0.05. We used Monte Carlo Sampling during 5000 iterations similar to [15], but in our case, Monte Carlo runs after a meta-heuristic complete execution. For this sampling the *path* was defined as 10. All meta-heuristics run for 500 generations, with the maximum population size of 100 for two and 210 for five objectives. For MOEA/DD we followed [14] and set $Nr = 1$, $\delta = 10$ and probability as 0.9. Regarding evolutionary operators, we followed [4], where the SBX Crossover and Polynomial Mutation were employed and setup with distribution set as $\{n_c = 20\}$ and $\{n_m = 20\}$ respectively with probabilities $\{p_c = 0.9\}$ and $\{p_m = 1/n_p\}$, where n_p is the number of decision variables in the problem. Regarding the reasoning engine, we performed the experiment considering 200 agents.

Implementation Tools: NOVA was coded using Java JDK 14 using jMetal 5.7 [18] and jMetalHyperHeuristicHelper [1].

We discuss our results from three different perspectives. First, Hypervolume and IGD+ averages are compared to understand the performance of different algorithms in this context. Secondly, we present how is the Pareto Front for each of the meta-heuristics. Finally, we analyse the best solutions from the problem perspective.

5.1 Hypervolume and IGD+ comparisons

We employed Hypervolume [13], and IGD+ [12] averages obtained from the 30 executions as the algorithms performance comparison criterion. This is necessary because MOEAs produce a set of non-dominated solutions, which makes each algorithm have a set of values to be compared. However, direct comparisons between solutions sets is difficult, therefore we need a single value that summarises the algorithm's performance. In Hypervolume, for example, we calculate the area (or volume) from each solution to a reference point. We treat them as points in the Cartesian space. For minimisation problems, we set this reference point as the worst possible. Thus, when algorithm A has a higher hypervolume value than algorithm B, it means that solutions from A are more distant to the worst point, and then the found Pareto Front provides more quality. For this purpose, first, for each problem (two and five objectives), we joined all results obtained by all algorithms, found the nadir point (worst found), necessary for Hypervolume calculation, and took the Non-dominated set in order to generate the *known Pareto Front (PF_{known})*, necessary for IGD+ calculation. Then, we calculated Hypervolume and IGD+ for each of the executions and generated averages for both quality indicators for each algorithm. Finally, we compared these averages using Kruskal-Wallis as the statistical test with a confidence level of 99%. In order to perform this, we first identified which algorithm has the best average according to the quality indicator, thus, all the other algorithms are compared to the best, generating a set of *p-values*. We define an algorithm tied statistically with the best when a given *p-value* is superior to the significance level of 0.01.

Table 1 presents a meta-heuristic comparison for the two and five objective problems. Here the mean for 30 executions, standard deviation (*std*), and *max* value among the executions are presented.

Regarding *max* for Hypervolume for the two-objective problem, SPEA2 found the highest value. However, it also has the highest *std*, which means this high value rarely occurs. NSGA-II has the best average, but when we consider both mean and std, we can see why MOEA/DD and SPEA2 results are statistically tied with NSGA-II. Regarding IGD+, also NSGA-II is the best algorithm, but this time being the best one considering *std*, *mean* and *max*. Finally, we can clearly see that all of these algorithms except MOMBI2 can be a good option for solving this two-objective problem.

For the five objective problems comparison, the scenario is completely different. MOMBI2 was a terrible algorithm for two objectives, but here it found the best *max* value regarding to Hypervolume. In terms of *std* and *mean* MOEA/DD is the best algorithm. These results made MOMBI2 and MOEA/DD the best algorithms considering Hypervolume. However, in terms of IGD+, MOEA/DD stands

[1] https://github.com/vinixnan/jMetalHyperHeuristicHelper

Table 1. Hypervolume and IGD+ averages for two and five objectives, highlighted values mean best results, bold values mean statistically tied results with the best. For Hypervolume, higher values are considered the best, while for IGD+ smaller is preferred.

		2 objectives		5 objectives	
	Metric	Hypervolume	IGD+	Hypervolume	IGD+
MOMBI2	mean	0.030525	1.242683	**0.378164**	0.086026
	std	0.090247	0.354033	0.022794	0.026758
	max	0.425940	1.973713	0.412411	0.137319
MOEA/DD	mean	**0.859607**	**0.103377**	**0.386071**	**0.039012**
	std	0.090352	0.068957	0.009275	0.004931
	max	0.974304	0.263218	0.402583	0.047243
NSGA-II	mean	**0.904303**	**0.056089**	0.321709	0.109361
	std	0.084679	0.053044	0.014935	0.014614
	max	0.999828	0.239230	0.353739	0.145614
SPEA2	mean	**0.862924**	**0.096932**	0.343156	0.075612
	std	0.180627	0.143008	0.009625	0.006417
	max	0.999379	0.704847	0.361828	0.091765

as the best algorithm with a better IGD+ average with a statistical difference. Moreover, it also had the smallest *std* and *max* values.

Figures 5 and 6 present, respectively, box-plots for Hypervolume and IGD+ for the two-objective problem. Basically, these figures represent visually the same results shown in Table 1. We can see how MOMBI2 is the algorithm with more variance (in Table 1 regarding *std*) while NSGA-II is the one with less variance. The reason for that is while NSGA-II usually found non-dominated solutions, MOMBI2 found dominated solutions making it sometimes having near to zero Hypervolume values. This is also clear when we analyse considering IGD+ where MOMBI2 always has bigger values.

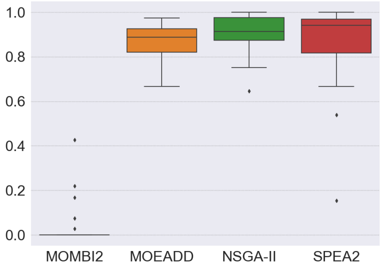

Figure 5. Hypervolume box-plot for two objectives

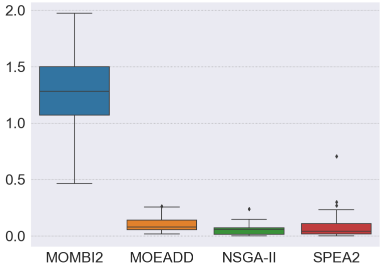

Figure 6. IGD+ box-plot for two objectives

Figures 7 and 8 present respectively box-plots for Hypervolume and IGD+ considering the five-objectives problem. Basically, these figures represent visually the same results shown in Table 1. Here we can see, for Hypervolume, how MOMBI2 and MOEA/DD perform better than NSGA-II and SPEA2. However, unlike MOEA/DD, MOMBI2 has a big variance having the biggest Hypervolume values (considering one single execution) and minimal values smaller than NSGA-II and SPEA2 maximum values. MOEA/DD is more stable in terms of results, even not having the maximum value among the algorithms, which is the best choice for this problem. This is even more clear when we take into consideration IGD+ values where MOEA/DD is clearly the best-performing algorithm in all aspects.

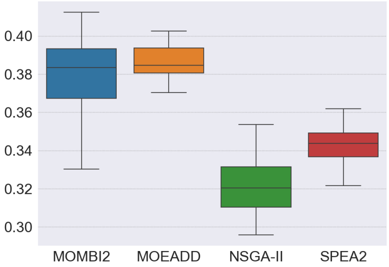

Figure 7. Hypervolume box-plot for five objectives

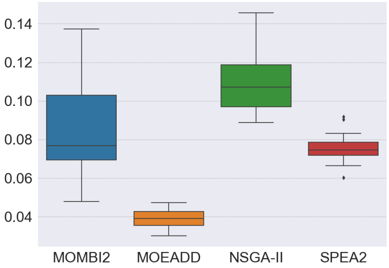

Figure 8. IGD+ box-plot for five objectives

5.2 *Pareto Fronts Analysis*

In this section, Pareto Fronts generated by each algorithm are compared. Interactive graphs can be found in **here**. Dominated solutions were also considered in order to provide a good view of how an algorithm can outperform others in terms of Pareto dominance. Fig. 9 presents this for the two objective problem. Here we can see why MOMBI2 had bad Hypervolume and IGD+ results in the previous section due to the fact most of the solutions are dominated. NSGA-II also has a good amount of solutions. However, several are non-dominated, which means this algorithm has good results according to quality indicators.

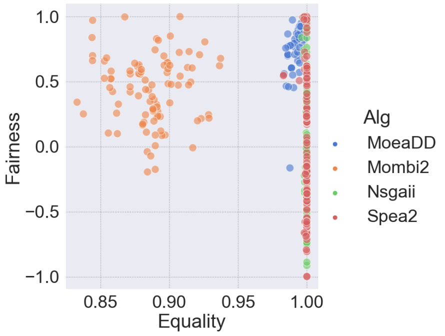

Figure 9. Pareto Front for algorithms at one execution considering two objectives

Table 2. NOVA best solution selected by the Reasoner engine for five objectives

Parameters	Fairness	Equality	Wealth	GainedAmount	CollectPortion
collect = [70.54%, 12.35%, 27.00%, 44.45%, 37.08%] redistribute= [99.79%, 18.99%, 89.39%, 77.21%, 86.50%] catch=96.05% fine=84.3634%	0.8	0.76	0.24	1.05	0.29

For the five-objective problem, we split the objectives into four groups by combining objectives one and two, which are used in the two-objective problem, with one of the other objectives. Because of space limits, we decided to plot only two fronts here, and other images can be seen at **this link**.

From Figure 10, that shows objectives 1, 2 and 3, we can view the Pareto shape, which is somehow linear and disconnected. NSGA-II has solutions on extremes while MOEAD/D is more spread. Figure 11 presents 3D plots for objectives 1, 2 and 4. Here we can see how MOEA/DD have solutions both spread and at extremes points. This is an excellent aspect of performance and corroborates what happened for Hypervolume values. We can also see MOMBI2 performing well here with values less spread but near to optimal. That is the reason why MOMBI2 got statistically tied results with MOEA/DD in Hypervolume, but the less diverse solutions made it have worse results when compared to MOEA/DD regarding IGD+.

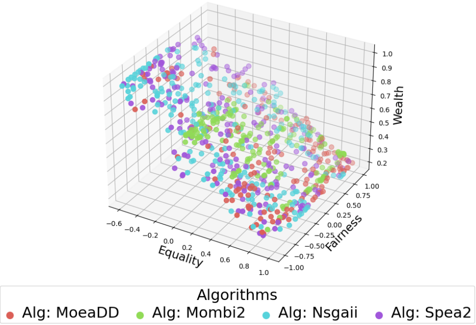

Figure 10. Pareto Front for algorithms for objectives 1, 2 and 3

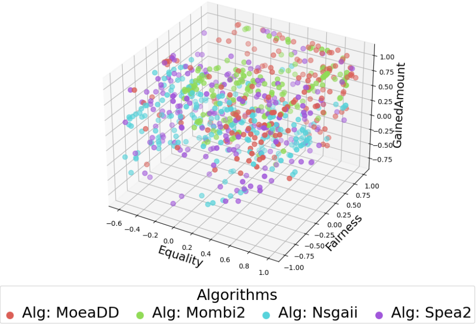

Figure 11. Pareto Front for algorithms for 1, 2 and 4 objectives

5.3 *Solution analysis*

In this section, we analyse the results taken after the reasoning engine runs. First, in Table 3, from comparing NOVA selected solution in case of the two-objectives scenario with the ones provided in [15], we can find that NOVA was able to achieve better results although it is aligning two values simultaneously unlike [15]. Second, in Table 2 we present the solution found by NOVA in the five-objective optimisation problem.

Discussion After reviewing the results, it can be noted that NOVA was successfully able to:

Table 3. Best two-objective solutions generated by NOVA according to the Reasoner compared against solutions provided in [15]

Parameters	Fairness	Equality
Montes and Sierra [15]		

collect = [20%, 29%, 26%, 35%, 27%]
redistribute = [20%, 22%, 19%, 26%, 13%]
catch = 44%
fine = 61%
	0	0.95

collect = [1%, 30%, 37%, 72%, 66%]
redistribute = [2%, 23%, 42%, 24%, 9%]
catch = 45%
fine = 56%
	0.93	0.59

NOVA		

collect = [78.45%, 51.92%, 60.63%, 56.13%, 63.15%]
redistribute= [52.75%, 69.74%, 47.87%, 51.03%, 54.30%]
catch= 91.72%
fine= 47.5155%
	0.80	0.86

collect = [1.96%, 16.38%, 35.72%, 25.76%, 36.19%]
redistribute= [51.82%, 43.57%, 52.13%, 58.69%, 63.23%]
catch= 99.88%
fine=22.1505%
	0	0.93

- **Optimise multiple values regardless of their compatibility:** NOVA was able to reach this by turning the values to objectives and formalising the problem as a MOP. The produced Pareto Front set ensures having the optimum solutions for the combination of all objectives regardless of their compatibility.
- **Select the best set of norms for heterogeneous groups of agents:** NOVA reached this by using the parameterised norms techniques that allow each group to have its own norms' values.
- **Align independent sets of values and norms:** by defining the norms set as the decision variables set in the MOP and the values set as the objectives to be optimised, NOVA separated the concept of norms and values and had independent sets.

6 Conclusion

In this paper, we proposed NOVA, a multi-value promotion model that uses multi-objective evolutionary algorithms to produce optimum parametric set of norms that is aligned with multiple simultaneously shared values by heterogeneous groups of agents. Moreover, it aligns values that may be incompatible such as fairness and equality, as ensuring fairness does not necessarily support equality. More importantly, we show how different algorithms can have different performance in one domain by analysing the performance of four evolutionary algorithms (NSGA-II, MOEA/DD, SPEA2, and MOMBI2). Furthermore, we enabled NOVA to produced final single optimum solution by using a decentralised reasoner.

As future work, we plan to develop an online multi-value alignment model, using a hyper-heuristic approach[7]. Subsequently, the limitation of only including a static and limited number of predefined norms will be addressed by the online norm emergence techniques.

References

[1] S. F. Adra, *Improving Convergence, Diversity and Pertinency in Multiobjective Optimisation*, Ph.D. dissertation, Department of Automatic Control and Systems Engineering, The University of Sheffield, 2007.

[2] Trevor Bench-Capon and Sanjay Modgil, 'Norms and value based reasoning: justifying compliance and violation', *Artificial Intelligence and Law*, **25**(1), 29–64, (2017).

[3] I. Boussaid, J. Lepagnot, and P. Siarry, 'A survey on optimization metaheuristics', *Information Sciences*, **237**, 82 – 117, (2013).

[4] L. Bradstreet, L. Barone, L. While, S. Huband, and P. Hingston, 'Use of the WFG toolkit and PISA for comparison of MOEAs', in *2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pp. 382–389, Honolulu, USA, (April 2007). IEEE.

[5] Duarte Coelho, Ana Madureira, Ivo Pereira, and Ramiro Gonçalves, 'A review on moea and metaheuristics for feature-selection', in *International Conference on Innovations in Bio-Inspired Computing and Applications*, pp. 216–225. Springer, (2021).

[6] Vinicius Renan de Carvalho and Fatemeh Golpayegani, 'Satisfying user preferences in optimised ridesharing services: A multi-agent multi-objective optimisation approach', *Applied Intelligence*, **52**(10), 11257–11272, (aug 2022).

[7] Vinicius Renan de Carvalho, Kate Larson, Anarosa Alves Franco Brandão, and Jaime Sim ao Sichman, 'Applying social choice theory to solve engineering multi-objective optimization problems', *Journal of Control, Automation and Electrical Systems (JCAE)*, **31**(6), 119–128, (February 2020).

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *IEEE Trans. Evol. Comput.*, **6**(2), 182–197, (Apr 2002).

[9] Corrado Giugni, *Variabilità e mutabilità: Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche*, Tipogr. di P. Cuppini, 1912.

[10] Samaneh Heidari, Nanda Wijermans, and Frank Dignum, 'Agents with dynamic social norms', in *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pp. 112–124. Springer, (2019).

[11] Raquel Hernández Gómez and Carlos A. Coello Coello, 'Improved metaheuristic based on the r2 indicator for many-objective optimization', in *Proc. of the 2015 Annual Conf. on Genetic and Evolutionary Computation*, GECCO '15, pp. 679–686, New York, NY, USA, (2015). ACM.

[12] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima, 'Modified distance calculation in generational distance and inverted generational distance', in *Evolutionary Multi-Criterion Optimization*, eds., António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, pp. 110–125, Berlin Heidelberg, (2015). Springer International Publishing.

[13] Andrzej Jaszkiewicz, 'Improved quick hypervolume algorithm', *Computers & Operations Research*, **90**, 72–83, (2018).

[14] Ke Li, Kalyanmoy Deb, Qingfu Zhang, and Sam Kwong, 'An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition', *IEEE Trans. Evol. Comput.*, **19**(5), 694–716, (oct 2015).

[15] Nieves Montes and Carles Sierra, *Value-Guided Synthesis of Parametric Normative Systems*, 907–915, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2021.

[16] Andreasa Morris-Martin, Marina De Vos, and Julian Padget, 'Norm emergence in multiagent systems: a viewpoint paper', *Autonomous Agents and Multi-Agent Systems*, **33**, 706–749, (2019).

[17] Muhammad Nagy, Yasser Mansour, and Sherif Abdelmohsen, 'Multi-objective optimization methods as a decision making strategy', *International Journal of Engineering Research and Technology*, **9**(3), (2020).

[18] A. J. Nebro, J. J. Durillo, and M. Vergne, 'Redesigning the jmetal multi-objective optimization framework', in *Proc. of the Companion Publication of the 2015 Annual Conf. on Genetic and Evolutionary Computation*, GECCO Companion '15, pp. 1093–1100, New York, NY, USA, (2015). ACM.

[19] Ritesh Noothigattu, Djallel Bouneffouf, Nicholas Mattei, Rachita Chandra, Piyush Madan, Kush R Varshney, Murray Campbell, Moninder Singh, and Francesca Rossi, 'Teaching ai agents ethical values using reinforcement learning and policy orchestration', *IBM Journal of Research and Development*, **63**(4/5), 2–1, (2019).

[20] Gabriela Ochoa, Lee A Christie, Alexander E Brownlee, and Andrew Hoyle, 'Multi-objective evolutionary design of antibiotic treatments', *Artificial intelligence in medicine*, **102**, 101759, (2020).

[21] Bo-Yang Qu, YS Zhu, YC Jiao, MY Wu, Ponnuthurai N Suganthan, and Jing J Liang, 'A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems', *Swarm and Evolutionary Computation*, **38**, 1–11, (2018).

[22] Maha Riad and Fatemeh Golpayegani, 'Run-time norms synthesis in multi-objective multi-agent systems', in *International Workshop on Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems*, pp. 78–93. Springer, (2022).

[23] Marc Serramia, Maite Lopez-Sanchez, and Juan A Rodriguez-Aguilar, 'A qualitative approach to composing value-aligned norm systems', in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1233–1241, (2020).

[24] Carles Sierra, Nardine Osman, Pablo Noriega, Jordi Sabater-Mir, and Antoni Perelló. Value alignment: a formal approach, 2021.

[25] JA Vázquez-Rodríguez and S Petrovic, 'A mixture experiments multi-objective hyper-heuristic', *Journal of the Operational Research Society*, **64**(11), 1664–1675, (2012).

[26] E. Zitzler, M. Laumanns, and L. Thiele, 'SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization', in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pp. 95–100. CIMNE, (2001).