Fair Division with Storage and an Application to Water Allocation

Eyal Briman[1], Nimrod Talmon[1], Stephane Airiau[2], Umberto Grandi [3], Jerome Lang [2], Jerome Mengin [3] and Faria Nasiri Mofakham[4]

[1]Ben Gurion University of the Negev
[2]Paris Dauphine University
[3]Toulouse University Capitole
[4]University of Isfahan

Abstract. We study a model of fair division that is motivated by the fair distribution of water. In our basic setting, there is a central scarce, time-varying source of water that can be stored (possibly partially); and several agents with different, time-varying demand of water. We consider several mechanisms for allocating water at each point in time, study their properties, and design efficient algorithms that achieve these goals for most cases, based on the maximization of utilitarian, egalitarian or Nash social welfare. In order to show the practical feasibility of our approach, we run our algorithms on simulated data as well as on simplified real data. The simulation results suggest (and perhaps confirm) that Nash social welfare is a sweet spot between efficiency and fairness.

1 Introduction

Water scarcity poses a significant challenge in many regions worldwide, necessitating the fair distribution of water among various agents in different sectors. Motivated by the fair distribution of water, in this paper we develop a fair division model for temporal resource allocation, and explore the application of fair division algorithms to manage water resources in regions with water shortages. We focus only on farmers with different sizes of fields growing different kinds of crops which require different amount of water. Indeed, ensuring equitable and efficient water distribution among diverse agents is of paramount importance, particularly given the significant variations in water needs among different users. The development of algorithms that can achieve fair allocation of water resources is crucial to address environmental sustainability, social equity, and economic stability. Such algorithms can optimize water distribution, considering factors like population density, agricultural demands, industrial requirements, and ecological preservation, thereby promoting responsible water management and safeguarding the fundamental right to access clean water for all individuals and sectors.

Beyond the equitable distribution of water, our proposed model may find application in other temporal resource allocation domains, provided they meet the following model requirements:

1. A divisible resource must be allocated among a set of agents at each point in a (discrete) set of time steps.
2. The resource can be stored centrally, with or without a limit. There may be an evaporation factor expressing that at each time point, a fraction of stored water will be lost.
3. At each point in time, an additional amount of resource comes in, following a predictable pattern.
4. Each agent has a known demand function specifying the amount of resource they require at each point in time. This demand function is established from the outset.

Requirement 1 places our study in the setting of *temporal fair division of a divisible resource*. Requirement 2 is mild and allows for a continuum between unlimited storage capacity to zero storage capacity, as well as a continuum between no evaporation at all and full evaporation (full evaporation being equivalent to no storage). Requirement 3 is an important simplification, as water income from rainfall is more of a stochastic nature. However, it makes sense to study this simplified version first, and we will argue later that our results can be adapted rather easily to stochastic water income. Requirement 4 is also a simplification, as demand can be stochastic too, but it is reasonable enough in domains such as agriculture or industry.

Informally speaking, in our model we have a centralized supplying entity, which possesses a predictable future supply curve in time; and an agent community, wherein each agent exhibits its unique desired demand curve over time – i.e., its demand over time. The objective is to efficiently allocate the available supply among these agents, with their utility determined by the fractional demand that is assigned to them. Essentially, we seek to strike a balance between optimizing individual agent utility and promoting fairness within the allocation process, while ensuring strategy-proofness. To achieve this, we consider the three classical notions of social welfare: utilitarian, Nash product, and egalitarian. While we consider water allocation as the primary application of our model, it also makes sense to other domains such as allocation of monetary resources (generally with full storage, except when some funds have a time limit), and electricity with storage, as new technologies are providing avenues for storing thermal energy at scale (used for electricity production) while minimizing environmental impact [3].

On the theoretical side, we show the existence of efficient algorithms that distribute the resource according to the different fairness notions described above, as well as achieve certain axiomatic properties. On the experimental side, we report on computer-based simulations done to identify the different behavior of these algorithms for simulated data ad simplified real-world data.

In this study, we consider a particular, simplified resource distribution model, recognizing that real-world systems encompass a myriad of additional complexities, notably distributed production and supply mechanisms. While we believe that our model serves as a valuable starting point for understanding fundamental principles in the context of fair division over time, incorporating further aspects, such as the distributed production and supply is an intriguing future direction for research.

Outline After discussing related work, we present our model of fair division with storage. Then we show how optimal allocations can be computed, exactly or approximately. We also discuss a few basic properties of the optimal solutions. Finally, we analyse the output of computer simulations, both on synthetic and on 'semi-real' datasets.

1.1 Related work

Fair allocation under Leontief preferences Given a set of m divisible resources, a Leontief utility function has the form

$$u(x_1, \ldots, x_m) = \max_{j \in \{1 \ldots m\}} \frac{x_j}{d(j)}$$

where $d(j)$ is the amount of resource j demanded by the agent and x_i is the amount of resource given to her. Fair division with Leontief utilities has received some attention, particularly in the context of fair division of computer resources: each resource j has a limited supply r_j, and each of n agents has a Leontief utility function u_i associated with a demand vector.

Dominant resource fairness (DRF) [9] assumes that every agent has a Leontief utility function with a positive demand for each resource. Given a demand vector d_i for agent i, the *normalized demand vector* d_i^* is defined as the fraction of the total supply of each resource demanded by the agent. For each agent, the resource maximizing normalized demand is called the dominant resource. Dominant resource fairness equalizes the share of the dominant resource given to all agents. Dominance resource fairness is egalitarian in essence: it maximizes egalitarian social welfare, provided the agents' utilities are normalized such that each agent's utility is the amount of their dominant resource that they get.

DRF is generalized by Parkes et al. [14] to account for the possibility for agents to have a zero demand on some resources (and also to weighted agents), and by Li and Xue [12], with generalized Leontief utilities, defined as the maximum between several Leontief utility functions. Codenotti and Varadarajan [7] show when agents have Leontief preferences, market equilibria (in a version of Fisher's model) can be computed in polynomial time. Yet another rule for fair allocation with Leontief utilities is the *no justified complaints* rule proposed by Dolev et al. [8].

These works regard the field of allocation of heterogeneous resources, with applications mostly to allocation of computer resources (especially, for cloud computing). The relation to our model is by identifying resources (in the DRF setting) with water to be allocated at time t in ours. However, none of these works consider storage (and for good reasons: bandwidth and CPU usage cannot be stored).

Repeated fair allocation Repeated fair allocation has received some attention in the context of *indivisible* goods and/or chores [4, 10]. A related line of work in dynamic fair division is *online* fair division, where indivisible items arriving at different time points have to be allocated as soon as they are available (see [2] for a survey). Yet another related line of related work is that of perpetual voting and related models [11, 5, 6], in which repeated decisions are being made, which relates – but is not identical – to the fact that we make a decision for each time step.

2 The model

2.1 Supply and demands

An *instance* of the formal model of water allocation we consider contains the following input ingredients:

- m time bins $T = \{t_1, \cdots, t_m\}$ representing some period of time;
- A *supply* function $S : t \to \mathbb{Q}^+$, where $t \in \{t_1, \cdots, t_m\}$ corresponds to the (additional) supply in time t of the good (which from now on we call water).
- A set of n agents, where agent $i \in [n]$ has its *demand* over T; this is denoted by d_i such that $d_i : t \to \mathbb{Q}^+$, where $i \in [n]$, $t \in \{t_1, \cdots, t_m\}$, corresponds to the demand of agent i in time period t. As a non-triviality assumption we assume that there is a point $t \in \{t_1, \cdots, t_m\}$ where the global demand is non-null. In some situations, it is appropriate to make the further assumption that demands are on intensities and not absolute quantities, i.e., that $\sum_{t \in \{t_1, \cdots, t_m\}} d_i(t) = k$ for each each agent $i \in [n]$, having k to be a constant. We will say that in this case the demands are *normalised*. Except when it is clearly stated, we do not make this assumption. Note that when distributing water among farms / fields, this assumption does not seem appropriate, as the amount of water that is needed depends on the crops and the field's size. This normalizing assumption may be relevant in other contexts, for instance when distributing energy, or computer resources, among agents that have the same stakes. This assumption is also reminiscent of cumulative voting, where a set of points need to be distributed over candidates.
- a common reservoir / storage facility of capacity $C \in \mathbb{Q}^+ \cup \{+\infty\}$. When $C = 0$ (respectively, $C = +\infty$), we say that there is no storage (respectively, full storage). We consider an evaporation factor $E : T \to [0, 1]$; it represents the percentage of residual water amount after evaporation during a single unit of time, dependent on the period of the year. In the following, we assume that the reservoir is initially empty. However, our analysis would remain the same if we assumed that the reservoir contains a quantity q_0 water initially with the additional constraint that the reservoir should contain at least q_0 at the end of the last time step. For simplicity, we assume $q_0 = 0$.

It is thus convenient to denote an instance of our water allocation model by (S, D, C, E), where $D = \{d_1, \ldots, d_n\}$. We write (S, D) for instances with no storage ($C = 0$, and E is irrelevant).

2.2 Allocations

Given an instance (S, D, C, E) of the model, a *solution* W corresponds to a division of the total available supply to the different agents. Formally:

- $W = \{w_1, \ldots, w_n\}$, where w_i, $i \in [n]$, is a function $w_i : t \to \mathbb{Q}^+$, where $w_i(t)$, $i \in [n]$, $t \in \{t_1, \cdots, t_m\}$, corresponding to the amount of supply assigned to agent i in time period t. We refer to $w_i(t)$ as the *water allocation* for agent i at time period t. Slightly abusing notation, it is convenient to denote by $W(t)$ the total allocation at time t: $W(t) = \sum_i w_i(t)$.
- $\mathbf{W}$ denotes the feasible space: the set of those allocation vectors W such that at every time point t, allocation $W(t)$ does not exceed supply $S(t)$ plus what is left in the storage after evaporation.

Formally, we introduce $R(t)$ to represent the quantity of water remaining in the reservoir at the beginning of time period t. Then we must have:

- $R(t_1) = 0$, $R(t_{i+1}) = E(t_i) \times \min(C, (R(t_i) + S(t_i) - W(t_i))$, and
- $W(t_i) \leq S(t_i) + R(t_i)$ at every time point t_i.

When there is no storage, $C = 0$, this reduces to: $W(t_i) \leq S(t_i)$ at every time point t_i.

2.3 Agent utility as fraction of demand

Given an instance (S, D, C, E) and a solution W, we define the *utility* of agent i – denoted by $util(i)$ (where (S, D, C, E) and W are clear from the context) – to be the largest value $\alpha \in \mathbb{Q}^+$ for which it holds that $w_i(t) \geq \alpha \cdot d_i(t)$, for each $t \in \{t_1, \cdots, t_m\}$. Formally, $util(i) = \min_{\{t|d_i(t)>0\}} \frac{w_i(t)}{d_i(t)}$. If $d_i(t) = 0$ for all t then agent i can be safely excluded from the model.

Our definition of utility captures the agricultural intuition that the agent can only "grow" a fraction equals to $util(i)$ of what it intended to do with the water, with the minimum allocation of water defining what "crop" survive. This is exactly the setting of Leontief utilities (cf. Section 1.1), with a different interpretation though, since here resources correspond to water used at different time points. This implies that our model *without storage* correspond to fair allocation with Leontief utilities. With storage, however, the constraints bearing on water consumption at different time points (and the importance of the direction of time) significantly departs from fair allocation with Leontief utilities. One simplification from a more realistic model is the absence of a maximum demand: if water was abundant, our model could allow to cultivate more than 100% of a field. However, our study is only interesting in the case where water is scarce, i.e. there is not enough water for the needs of all the farmers.

2.4 Tight allocations

Following the definition of agent utility above, and for regularization reasons, we will be interested only in a specific sub-class of water allocations, as defined next.

Definition 1. *A solution W to a water allocation instance (S, D, C, E) is* tight *if $w_i(t) = util(i) \cdot d_i(t)$ for all agents and all $t \in \{t_1, \cdots, t_m\}$.*

Thus, when restricting to tight solutions, we will use α_i to denote the water allocation $w_i(t) = \alpha_i d_i(t)$. Informally, a tight water allocation allocates a constant fraction of the demand of water to an agent. In doing so, it corresponds to a tightness requirement in which it never gives more water to an agent than the agent needs (given the water it gets at other times). Given our definition of utility, without loss of generality we can from now on consider only tight allocations. Without storage, the water that is not used is lost: in the irrigation application, it would not be useful for the crop. Technically, such unused water could be added to the tight allocation: the resulting allocation would no longer tight, but the utility of each agent would remain the same. With storage, the unused water is of course the water stored and is available for coming time steps.

2.5 Optimization criteria as social welfare measures

Given the definition of the agent utility as given above, we consider several, different optimization/fairness goals. The utilitarian criterion is quite direct; it has the positive property of maximizing social welfare, however, it can completely ignore many agents. In the ***utilitarian*** version of the problem we aim at maximizing the sum of utilities; i.e., we are looking for the following:

$$\arg\max_W \left(\sum_{i \in [n]} util(i) \right) .$$

The ***egalitarian*** social welfare is quite extreme; on one hand, it does not ignore agents, however, on the other hand, it may reduce the social welfare significantly just for a single agent. In this version of the problem we aim maximizing the minimum utility; i.e., we are looking for the following:

$$\arg\max_W \left(\min_{i \in [n]} util(i) \right) .$$

Maximizing *egalitarian* social welfare without storage is almost equivalent to dominance resource fairness (DRF), up to normalization: in DRF, an agent's utility is the share of their dominant resource. This can be written as maximum egalitarian social welfare as in our setting, provided that demands are re-normalized in such a way that each agent has a unit demand for their dominant resource.

The ***prioritarian*** social welfare is sometimes considered as a trade-off between the utilitarian the egalitarian one. We model this by the Nash product. Hence, we aim at maximizing the multiplication of the utilities; i.e., we are looking for the following:

$$\arg\max_W \left(\prod_{i \in [n]} util(i) \right) .$$

3 Computation and properties

We design algorithms to compute water allocations according to the three optimisation criteria considered, and investigate their theoretical properties of strategy-proofness and Pareto-efficiency.

3.1 Computing optimal allocations

We start with the egalitarian solution, for which we are able to find a close form solution.

Theorem 1. *Every water allocation instance without storage has a tight egalitarian solution that allocates $w_i(t) = \alpha \cdot d_i(t)$ to each i, where*

$$\alpha = \frac{S(t^*)}{\sum_i d_i(t^*)} \text{ and } t^* \in \arg\min_{\{t \in \{t_1, \cdots, t_m\} | \sum_i d_i(t) > 0\}} \frac{S(t)}{\sum_i d_i(t)}.$$

This solution is unique and can be computed in polynomial time.

Proof. Time point $t^* \in \arg\min_{\{t \in \{t_1, \cdots, t_m\} | \sum_i d_i(t) > 0\}} \frac{S(t)}{\sum_i d_i(t)}$ is a time period in which the total water demand (i.e the sum of all agents' demands) reaches its peak, i.e., α is the fraction of water that can be allocated equally to each agent in the time of peak demand. Consider the tight solution W^* that gives utility α to every agent: $w^*_i(t) = \alpha d_i(t)$ for every agent i at every time point t. Consider another solution W' such that $w'_i(t) \geq util_{w'}(i) \times d_i(t)$ for every agent i and time point t. In order to beat W^* on the egalitarian criterion, W' must give a strictly better utility than W^* for every agent i: $\alpha < util_{w'}(i)$. Since $\frac{S(t)}{\sum_i d_i(t)}$ is minimum at time t^*, $\sum_i d_i(t^*) \neq 0$ (recall that we assume that demands are non null and ≥ 0, so there is at least one time point t where $\sum_i d_i(t) \neq 0$). Then we must have $\sum_i \alpha d_i(t^*) < \sum_i util_{w'}(i) d_i(t^*) = \sum_i w'_i(t^*) \leq S(t^*) = \alpha \sum_i d_i(t^*)$, which is a contradiction. Therefore W^* is the only tight solution that maximizes the egalitarian criterion. □

The result also holds when storage is available, although the close form of α is more complicated.

Theorem 2. *Every water allocation instance with storage has a tight egalitarian solution that allocates $w_i = \alpha \cdot d_i$ to each i. This solution is unique and can be computed in polynomial time.*

Proof. We can use an iterative argument following the proof of Theorem 1. We can initially run the algorithm with no storage: we obtain a fist allocation. Let σ be the permutation such that $\frac{S(\sigma(t))}{\sum_i d_i(\sigma(t))}$ is ordered from the smallest to the largest. As long as there is enough available water in storage, we use a quantity of water r_l and split it between the first $l-1$ periods so that the following first $l+1$ terms are equal : $\frac{S(\sigma(1))+r_1^1+r_2^1+r_3^1\cdots+r_l^1}{\sum_i d_i(\sigma(1))} = \frac{S(\sigma(2))+r_2^2+\cdots+r_l^2}{\sum_i d_i(\sigma(2))} = \cdots = \frac{S(\sigma(l))+r_l^l l}{\sum_i d_i(\sigma(l))} = \frac{S(\sigma(l+1))}{\sum_i d_i(\sigma(l+1))}$. Informally, we raise the level of the first peak to the second peak at the first iteration, then (if the available water is sufficient), we can raise the level of the first two peaks to the level of the third peak and so on. We can use a simple linear program to share the available water. Two cases may occur in the process. Either there is a point l where there is not enough available water to reach the equality of the first l terms. In this case, the algorithm stops, and we have an egalitarian solution by sharing equally the water. The water left cannot be used to improve the utility of any agent. If there is water available, all periods have been served, and we can add water to all periods to improve the utility of each agent, and we obtain $\frac{S(\sigma(1))+r_1^1+r_2^1+r_3^1\cdots+r_T^1}{\sum_i d_i(\sigma(1))} = \frac{S(\sigma(2))+r_2^2+\cdots+r_l^2+r_T^2}{\sum_i d_i(\sigma(2))} = \cdots = \frac{S(\sigma(l))+r_{T-1}^{T-1}+r_T^{T-1}}{\sum_i d_i(\sigma(l))} = \frac{S(\sigma(T))+r_T^T}{\sum_i d_i(\sigma(T))}$ and no water is left. In that case, sharing equally the water is egalitarian. Note that the maximum number of iteration is the number of periods, so the computational complexity is polynomial. □

Whether storage is available or not, our proofs show that all agents share the same α. Combined with our notion of utility, this implies that the utility of all agents is the same for the egalitarian solution: the egalitarian solution is actually stronger, it is equitable.

Corollary 1. *A tight egalitarian solution is equitable (i.e., the utility of each agent is the same) with or without storage.*

We now move to the case of the utilitarian solution, which can be computed by a linear program.

Theorem 3. *The utilitarian solution with and without storage can be computed in polynomial time using linear programming.*

Proof. For the case of storage, the utilitarian solution can be computed by using the following linear program:

Constants : $S(t)$ (supply in time $t \in \{t_1, \cdots, t_m\}$), capacity C, evaporation constant $E(t)$ at time $t \in \{t_1, \cdots, t_m\}$, and $d_i(t)$ (demand of agent i in time $t \in \{t_1, \cdots, t_m\}$), $i \in [n]$, $t \in \{t_1, \cdots, t_m\}$.

Real-valued variables $\alpha_i, w_i(t), X(t)$ $i \in [n], t \in \{t_1, \cdots, t_m\}$. note that $X(t)$ is the amount of water stored on time t.

$$\text{Maximize} \quad \sum_{i=1}^{n} \alpha_i.$$

Subject to

$$w_i(t) \geq \alpha_i \cdot d_i(t) \quad , \forall i \in [n], t \in \{t_1, \cdots, t_m\} \text{ (tightness)}$$

$$W(t) = \sum_{i=1}^{n} w_i(t)$$

$$W(t_1) \leq S(t_1)$$

$$W(t) \leq S(t) + X(t), \text{ for all } \in t > 1$$

$$X(t) \leq (X(t-1) + S(t-1) - W(t-1)) \cdot E(t), \quad \text{for all } t \in \{t_1, \cdots, t_m\}$$

$$X(t) \leq C, \forall t \in \{t_1, \cdots, t_m\}$$

□

Note that by adding a variable α_{min} and a constraint $\alpha_{min} \leq \alpha_i$ for each agent i, we can use the linear program in the proof of Theorem 3 to compute the egalitarian solution by changing the objective function to maximising α_{min}.

Because utilitarian and egalitarian social welfare can be formulated as linear programs, there always exists an optimal solution where the amount of water allocated to each agent at each time point is a rational number. This is no longer the case with Nash social welfare (which is something quite classical with maximal Nash welfare). As a consequence, assessing the computational complexity of the exact optimization problem is not what should be focused on. On the other hand, obtaining an approximate solution by convex programming is convenient. In our simulations for the Nash algorithm, we use Gurobi's log approximation[1], enabling us to set the objective function of the Nash LP formulation to be: Maximize $\sum_{i=1}^{n} \log(\alpha_i)$.

Example 1. *Consider the example in Table 1, modelling the demands of 3 agents over 3 time steps and the supply over 3 time steps. The (different) solutions to this water allocation problem obtained using the three proposed optimization criteria (without assuming storage) are presented in Table 2. If we consider a storage of 20 water units then the results are presented in Table 3. In both Tables 2 and 3, each α value is rounded up to two decimal places. In this Example we see that incorporating storage increases the social welfare and that utilitarian seems to be the most "unfair".*

Table 1. Agent demand and supply.

	$d_i(t_1)$	$d_i(t_2)$	$d_i(t_3)$
Agent 1	18.44	8.43	73.13
Agent 2	46.22	10.47	43.32
Agent 3	28.24	54.96	16.79
Supply	67	51	71

3.2 Pareto-efficiency and strategy-proofness

Pareto Efficiency means that there is no other solution for which all agents are at least as happy and at least one agent is strictly happier. Formally:

[1] https://www.gurobi.com/documentation/9.1/refman/py_model_agc_log.html

Table 2. Water allocation to agents with no storage.

Criterion	Agent	$w_i(t_1)$	$w_i(t_2)$	$w_i(t_3)$	$\lceil \boldsymbol{\alpha} \rceil$
Utilitarian	Agent 1	4.99	2.284	19.79	0.27
	Agent 2	41.84	9.474	39.22	0.91
	Agent 3	20.16	39.24	11.99	0.71
Egalitarian	Agent 1	9.83	4.49	38.96	0.53
	Agent 2	24.63	5.58	23.08	0.53
	Agent 3	15.05	29.29	8.95	0.53
Nash	Agent 1	7.65	3.5	30.33	0.41
	Agent 2	30.13	6.82	28.23	0.65
	Agent 3	20.91	40.68	12.43	0.74

Table 3. Water allocation to agents with 20 units of water storage.

Criterion	Agent	$w_i(t_1)$	$w_i(t_2)$	$w_i(t_3)$	$\lceil \boldsymbol{\alpha} \rceil$
Utilitarian	Agent 1	17.37	7.94	68.86	0.94
	Agent 2	21.39	4.84	20.05	0.46
	Agent 3	28.24	54.96	16.79	1.00
Egalitarian	Agent 1	13.30	6.08	52.74	0.72
	Agent 2	33.33	7.55	31.24	0.72
	Agent 3	20.37	39.64	12.11	0.72
Nash	Agent 1	18.44	8.43	73.12	1.00
	Agent 2	24.48	5.54	22.94	0.53
	Agent 3	24.08	46.86	14.32	0.85

Definition 2. *An algorithm $\mathcal{A}$ is* Pareto efficient *if, for each instance – denoting the utility of the agents given $\mathcal{A}$ by α – there is no other solution W' that corresponds to agent utilities α' such that $\alpha'_i \geq \alpha_i$, $i \in [n]$ and there is at least one i such that $\alpha'_i > \alpha_i$.*

Intuitively, Utilitarian, and Prioritarian satisfy Pareto efficiency, because a solution that only improves the agent utilities also achieves higher score; but, for Egalitarian, it does not hold because a solution violating the Pareto Efficiency may achieve the same Egalitarian score. The proof is included in the supplementary material.

Theorem 4. *The Utilitarian, and Prioritarian variants satisfy Pareto efficiency, but the Egalitarian variant does not satisfy Pareto Efficiency, even in the model without storage.*

We now turn to analyse the incentive structure of our proposed allocation mechanism. Recall that a mechanism is strategy-proof if agents have no incentive to misreport their demand, i.e., given a water allocation instance (S, D), for all $d'_i \neq d_i$ the utility of agent i in (S, D) is bigger or equal than the utility of i in (S, D') where D' is the profile of demands D where agent i reports d' instead of d. We start with egalitarian for which we have positive results with and without storage. We provide a formal proof for the non-storage case and a sketch for the storage case.

Theorem 5. *The egalitarian solution is strategy-proof for tight allocations, assuming no storage and the manipulator does not change the sum of her demands.*

Proof. Let (S, D) be an initial instance, and let W be its associated egalitarian solution. Consider agent i_0 who aims at changing the result in its favour by reporting a different demand function d'_{i_0}. Let (S, D') where $D' = (d_1, \ldots, d'_{i_0}, \ldots d_n)$ be the "manipulated" instance of the problem and W' be its associated egalitarian solution.

Let $T^* = \arg\min_{t \in \{t_1, \cdots, t_m\}} \frac{S(t)}{\sum_i d_i(t)}$ the set of peaks in the initial instance, and let $T' = \arg\min_{t \in \{t_1, \cdots, t_m\}} \frac{S(t)}{\sum_i d'_i(t)}$ be the new set of peaks in the manipulated instance. The egalitarian allocations are computed with $\alpha = \frac{S(t^*)}{\sum_i d_i(t^*)}$ for any $t^* \in T^*$ in the initial instance, and $\alpha' = \frac{S(t')}{\sum_i d_i(t')}$ for any $t' \in T'$ in the manipulated one.

Let $util_{i_0}$ denote the utility of agent i_0 with the initial allocation, and $util'_{i_0}$ her utility in the allocation made on the basis of the manipulated instance. Then $util_{i_0} = \alpha$ and $util'_{i_0} = \min_t \left(\frac{\alpha' d'_{i_0}(t)}{d_{i_0}(t)}\right) = \alpha' \min_t \frac{d'_{i_0}(t)}{d_{i_0}(t)}$. The manipulation is profitable if and only if $\alpha < \alpha' \min_t \frac{d'_{i_0}(t)}{d_{i_0}(t)}$. We distinguish two cases:

1) If $\alpha \geq \alpha'$: note that $\min_t \frac{d'_{i_0}(t)}{d_{i_0}(t)} < 1$, because i_0's modified demand must still sum to K, and there must be some time point t such that $d'_{i_0}(t) \neq d_{i_0}(t)$, so there must be some time point t such that $d'_{i_0}(t) < d_{i_0}(t)$. Therefore, in this case, $\alpha \geq \alpha' > \alpha' \min_t \frac{d'_{i_0}(t)}{d_{i_0}(t)}$, so the manipulation is not profitable.

2) If $\alpha < \alpha'$: recall that $W(t^*)$ (respectively $W'(t^*)$) is the total amount of water allocated at time t^* with the non-manipulated (resp. manipulated) instance. Let $\dot{D}(t^*)$ be the sum of the demands by the other agents at t^*, unchanged in the manipulated instance: $\dot{D}(t^*) = \sum_{i \neq i_0} d_i(t^*) = \sum_{i \neq i_0} d'_i(t^*)$. Then $W(t^*) = \alpha \dot{D}(t^*) + \alpha d_{i_0}(t^*)$ and $W'(t^*) = \alpha' \dot{D}(t^*) + \alpha' d'_{i_0}(t^*) > \alpha \dot{D}(t^*) + \alpha' d'_{i_0}(t^*)$ since $\alpha < \alpha'$. Moreover, $W'(t^*) \leq S(t^*)$ (allocation cannot exceed supply without storage), and $W(t^*) = S(t^*)$ since all water is allocated at time t^* in the non-manipulated instance, thus $W(t^*) \geq W'(t^*)$. This implies that $\alpha d_{i_0}(t^*) > \alpha' d'_{i_0}(t^*)$. It cannot be the case that $d_{i_0}(t^*) = 0$ since $\alpha' d'_{i_0}(t^*) \geq 0$, thus $\alpha > \alpha' \frac{d'_{i_0}(t^*)}{d_{i_0}(t^*)} \geq \alpha' \min_t \frac{d'_{i_0}(t)}{d_{i_0}(t)}$, which implies that the manipulation is not profitable.

In both cases above, the manipulation is not profitable. Therefore the egalitarian solution is strategy-proof. □

Theorem 6. *The egalitarian solution is strategy-proof for tight allocations, assuming storage and the manipulator does not change the sum of her demands.*

Proof. We re-use the notation of the previous proof. In an egalitarian solution, all agents share the same α and there must be a time step $t^\star$ for which no more water is available to increase the utility. As noted in the previous proof, it must be the case that $\alpha' > \alpha$. As a consequence, all other agents get more water as their demands remain the same (recall that in a tight allocation, an agent i gets $\alpha' d_i(t)$). To increase her utility, the manipulator must also get more water, so all agents strictly increase their water allocation. However as no water is available at $t^\star$, there is no additional water available, so the solution is strategy-proof. □

The manipulations considered assume that the manipulator i_0 does not change the sum of her demands, i.e. $\sum_t d_{i0}(t) = \sum_t d'_{i_0}(t)$. This assumption makes some sense in our irrigation application: the restriction requires the farmer to request the same total intensity of water, but she can change how she distribute it over the different time steps. Without such assumption, the following examples show that it is possible to manipulate[2].

Example 2. *Consider an instance with two time steps and two agents; let $S = (1, 1)$, $d_1 = (1/2, 1/2)$, $d_2 = (1, 0)$. the Egalitarian allocation is $\alpha = 1/(1 + 1/2) = 2/3$; Suppose now that agent 1 manipulates with demands $d'_1 = (1, 1)$: both time steps see the same overall demand $1 + 1$, and the Egalitarian allocation is $\alpha = 1/(1 + 1) = 1/2$; since both agents have the same demands, they both get the allocation $w' = (1, 1)/2 = (1/2, 1/2)$; for agent*

[2] Observe that dominant resource fairness is also strategyproof [9] and also relies on a normalization (albeit a different one).

2, this makes a utility of $1/2$*; but for the first agent, the utility, computed w.r.t. her true demands, is* $\min(\frac{1}{2}/\frac{1}{2}, \frac{1}{2}/\frac{1}{2}) = 1$.

The next results show that the utilitarian and prioritarian solutions are not strategy-proof.

Theorem 7. *Under tightness assumptions, the utilitarian solution is not strategy-proof, even without storage.*

Proof. Let us consider the following counterexample illustrated in Table 4. Under tightness assumptions, the utilitarian solution allocates the supply as shown.

Table 4. Utilitarian manipulation: agent true demands and allocation

Agent i	$d_i(t_1)$	$d_i(t_2)$	$d_i(t_3)$	$w_i(t_1)$	$w_i(t_2)$	$w_i(t_3)$
Agent 1	50	90	10	0	0	0
Agent 2	25	90	35	25	90	35
Agent 3	31	90	29	0	0	0
Supply	30	90	1000			

Considering the tightness constraint, this results in $\sum_{i=1}^{3} \alpha_i = 0 + 1 + 0 = 1$. If Agent 3 had declared his demand in: time step 1 as 25, time step 2 as 45, and time step 3 as 80, the allocation as shown in Table 5, would have resulted in an optimization of: $\alpha_1 + \alpha_2 + \alpha_3' = 0 + 1 + 0.2 = 1.2$, having α_3' to be the the untruthful α of agent 3. Then, the actual α of agent 3 would be 0.5, which is indeed larger than his original α of 0, and hence the manipulation works. □

Table 5. Utilitarian manipulation: allocation after manipulation.

Agent i	$w_i(t_1)$	$w_i(t_2)$	$w_i(t_3)$
Agent 1	0	0	0
Agent 2	5	18	7
Agent 3	25	45	80

Theorem 8. *Under tightness assumptions, the prioritarian solution is not strategy-proof, even without storage.*

Proof. Consider the counterexample illustrated in Table 6. Under tightness assumption, the utilitarian solution allocates the supply as shown.

Table 6. Prioritarian manipulation: agent true demands and allocation.

Agent i	$d_i(t_1)$	$d_i(t_2)$	$d_i(t_3)$	$w_i(t_1)$	$w_i(t_2)$	$w_i(t_3)$
Agent 1	50	90	10	11.67	21.00	2.33
Agent 2	25	90	35	11.90	42.85	16.66
Agent 3	30	90	30	11.43	34.29	11.43
Supply	35	120	1000			

Considering the tightness constraint, this results in $\prod_{i=1}^{3} \alpha_i = 0.0424$. If Agent 3 had declared his demand in: time step 1 as 25, time step 2 as 90, and time step 3 as 35, the allocation as shown in table 7, would have resulted in an optimization of: $\alpha_1 \cdot \alpha_2 \cdot \alpha_3' = 0.0506$, having α_3' to be the the untruthful α of agent 3. This means the actual α of agent 3 would be 0.398, which is indeed larger than his original α of 0.381, and hence the manipulation works. □

4 Computer-based simulations

In this section we report on computer-based simulations that explore the underlying structure of water allocation solutions provided by our algorithms, and the effect of different instance properties on the solutions. We describe the experimental setup, the evaluated algorithms, the datasets used, and the relevant parameters and measures.

Table 7. Water allocation to agents and supply after manipulation.

Agent i	$w_i(t_1)$	$w_i(t_2)$	$w_i(t_3)$
Agent 1	11.67	21.00	2.33
Agent 2	11.39	41.01	15.95
Agent 3	11.94	42.98	16.71
Supply	35	120	1000

4.1 Algorithms

We conducted experiments with the utilitarian, prioritarian, and egalitarian algorithms (using their algorithms as described above); as well as a further algorithm that we use as a baseline. We refer to this algorithm as the *equal allocation* algorithm.

Definition 3 (Equal allocation)**.** *The* equal allocation *algorithm operates as follows: for each agent, it assigns* $1/n$ *of the total supply in each time step, as well as* $1/n$ *of the storage capacity; then, it optimally solves each agent separately.*

Remark 1. *Note that we consider the equal allocation as a baseline. In particular, in a way, it does not foster any "communication" (or, central management) between the agents. Note also that it trivially satisfies Strategyproofness, Individual Rationality, and Envyfreeness. As for computation, it can be solved by linear programming (e.g., by using the utilitarian linear program for each agent separately; this is how we have computed it in the simulations).*

4.2 Datasets

We used two datasets: one that is entirely artificial and another that is based on real-world attributes. We explain their generation.

Semi-real data We conducted a simulation based on real data from the Western Negev area of Israel, focusing on 1381 fields irrigated with greywater (recycled sewage water) over a 6-month period from April to September. The data [1] includes details on the crops grown in each field and the field sizes measured in Dunams (equivalent to 1000 square meters). In our simulation, we examined water demand for fields cultivating wheat, maize, pumpkin, and potato. To estimate water requirements, we utilized a modified version of a table from a study by [13] adjusted for Dunams, presented in Table 8.

For each field, we calculated the daily water requirement by multiplying the crop-specific water requirement by 30 (assuming an average of 30 days per month) and then by the field size. This process generated a matrix with 1381 rows (representing each field) and 6 columns (for the 6 months of the study period).

Table 8. Crop Production (m³/dunam/day).

Crops	**Months**					
	IV	V	VI	VII	VIII	IX
Maize	1.4	2.6	3.4	3.9	3.6	2.2
Potato	1.6	2.5	3.9	4.2	3.3	1.6
Wheat	2.3	3.7	3.7	3.3	1.5	0.001
Pumpkin	0.001	3.2	3.9	4.1	3.7	2.3

As a post-processing, we normalized the data: the demands were normalized by dividing each value by the maximum demand observed across all fields and time. Then, the softmax function was applied.[3] This procedure ensured that the sum of demands over all

[3] $\text{softmax}(\textbf{demand}[i,t]) = \frac{e^{\textbf{demand}[i,t]}}{\sum_{t=1}^{k} e^{\textbf{demand}[i,t]}}$.

time steps for each field is equal but distributed differently. The supply was generated using a Dirichlet distribution[4] with $\alpha^6 = 1^6$, multiplied by a value generated uniformly at random between 10 and 40, and then multiplied by the number of fields, with 1 added to ensure all values are strictly greater than zero. The evaporation constant was also generated uniformly at random between 0 and 0.1.

Artificial data We also generated data that is completely artificial. In particular, with 500 agents (i.e., fields). The demand of each field in every instance was generated using a Dirichlet distribution with $\alpha^k = 1^k$, where k is the number of time steps (in our case, 12 months). We then multiplied all demands by 1000 and added 1 to each value, to ensure all values are strictly greater than zero. The demands can be presented as a matrix sized 500×12, where each row $i \in [500]$ represents the demand of field i.

The supply was generated using the same Dirichlet distribution, but multiplying the values by a uniformly random number between 500 and 1000. Furthermore, we multiplied the supply by 500 (the number of fields) and added 1 to all supply values to ensure all values are strictly greater than zero.

4.3 Measures

Given an instance I and some optimization goal F we arrive to a solution W – that corresponds to some agent satisfaction vector $\alpha_1, \ldots, \alpha_n$. We measure the quality of W using two measures:

- **Mean α:** This is the average of the agent satisfaction from the solution W – i.e., $\sum_{i\in[n]} \frac{\alpha_i}{n}$; note that it corresponds to the social welfare (i.e., to what the utilitarian solution directly optimizes).
- **Equality score:** This is the ratio between the minimum α and the maximum α; i.e., ${\min_{i\in[n]} \alpha_i}/{\max_{i\in[n]} \alpha_i}$; we view this as a fairness measure with equality score 1 being achieved from the egalitarian solution.

We use two measures regarding the inner instance structure:

- The average ℓ_1 distance between demands of fields (i.e., rows in the matrix). This helps us understand how demands of different fields are distributed differently, i.e., how similar are the agents in their demands. Results using this measure are provided in the supplementary material due to space constraints.
- Most "tight" allocation (peak)-Calculated as: $\arg\min_{t\in\{t_1,\cdots,t_m\}} \frac{Supply[t]}{\sum_{i=1}^{n} demands[i,t]}$; i.e., the minimal ratio of the peak (the most tight time step, as a measure of how constrained the instance is).

4.4 Experimental design

Both kind of simulations share common characteristics:

- There are n fields, each dedicated to growing one type of annual crop (plants that complete their life cycle within one year). A simplifying assumption in these simulations is that the yearly water demand (i.e., the sum of demands across all months) is equal for all fields. In reality, fields may vary in the types of crops grown and their corresponding water requirements, and also considering the different sizes of fields.
- There is a monthly water supply, and a centralized storage unit that exists with a maximum capacity C. Specifically, we examine the following scenarios: $C = 0$: no storage available; $C \neq \infty$: limited storage capacity; and $C = \infty$: unlimited storage capacity.

For both experiments, we tested capacity limits of 0, 50, 100, 200, and ∞, with the evaporation constant generated uniformly at random between 0 and 0.1. For the artificial data simulation, we conducted 100 repetitions of generating artificial data for each capacity limit. Each algorithm was implemented using Gurobi's Python package.

4.5 Results and Discussion

Figure 1 illustrates that the mean α increases as the storage capacity increases, in the simulation conducted with artificial data. This observation is further supported by a T-test with Bonferroni correction conducted between all pairs of capacities. Figure 1 demonstrates that the utilitarian algorithm receives the highest mean value of α, followed by Nash, then Egalitarian, and finally, the equal distribution mechanism. Additional figures presented in the supplementary material show less unambiguous differences between the capacities from the perspective of the equality measure. For instance, again on artificial data we can observe the egalitarian criterion has the highest equality measure, followed by Nash, the baseline algorithm, and then the utilitarian. Additional figures using semi-real data and the artificial data can be found in the supplementary material.

These results led us to the following conclusions: (1) Using coordination is better than no coordination at all, because Utilitarian, Nash, and Egalitarian are better than the decentralized no-mechanism (Equal Allocation); (2) There is indeed a visible trade off between social welfare and fairness; (3) Incorporating storage, increases the social welfare and the equality for all algorithms (as one could expect); (4) Nash is perhaps a sweet spot as it gains in equality but doesn't lose much in social welfare; and (5) In semi real world data simulations the differences between the algorithms are less visible.

5 Outlook

Motivated by various usecases, including water management, we have considered a model of fair division with storage. We have described different optimization criteria and showed the tradeoff between social welfare and fairness, using both a theoretical analysis and computer-based simulations.

In the context of water management, some real-world aspects are not captured by our model. These are interesting avenues for future research, and include: different importance weights for different agents; the existence of several water sources and the geographic and topological structure of the sources and the agents; the existence of different types of water (e.g., grey water); the inability of accurately forecasting rainfalls, necessitating the development of approximation algorithms that are built upon predictions. Also, our model can prove useful in related applications of continuous divisible goods over time such as money and energy management.

References

[1] https://data1-moag.opendata.arcgis.com/datasets/f2cbce5354024da28f93788c53b182d2_0/explore?location=31.524637%2C34.978485%2C8.87. Accessed on April 24, 2024.

[2] M. Aleksandrov and T. Walsh. Online fair division: A survey. In *AAAI*, pages 13557–13562. AAAI Press, 2020.

[4] $f(\mathbf{p}|\alpha) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} p_i^{\alpha_i - 1}$.

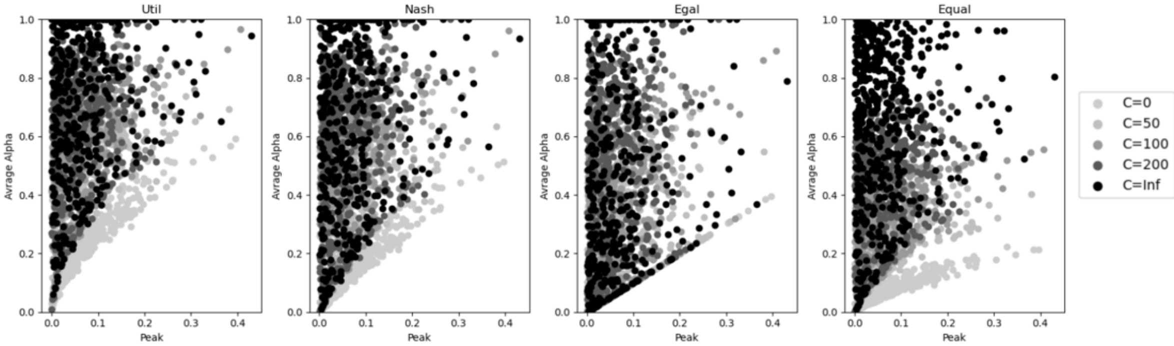

Figure 1. Average α as a function of peak - for all 4 algorithms, 5 capacities of storage and all 500 artificial instances generated.

[3] H. M. Ali, T.-u. Rehman, M. Arıcı, Z. Said, B. Duraković, H. I. Mohammed, R. Kumar, M. K. Rathod, O. Buyukdagli, and M. Teggar. Advances in thermal energy storage: Fundamentals and applications. *Progress in Energy and Combustion Science*, 100:101109, 2024.

[4] G. C. Balan, D. Richards, and S. Luke. Long-term fairness with bounded worst-case losses. *Auton. Agents Multi Agent Syst.*, 22(1):43–63, 2011.

[5] L. Bulteau, N. Hazon, R. Page, A. Rosenfeld, and N. Talmon. Justified representation for perpetual voting. *IEEE Access*, 9:96598–96612, 2021.

[6] N. Chandak, S. Goel, and D. Peters. Proportional aggregation of preferences for sequential decision making. In M. J. Wooldridge, J. G. Dy, and S. Natarajan, editors, *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[7] B. Codenotti and K. R. Varadarajan. Efficient computation of equilibrium prices for markets with Leontief utilities. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, 2004.

[8] D. Dolev, D. G. Feitelson, J. Y. Halpern, R. Kupferman, and N. Linial. No justified complaints: on fair sharing of multiple resources. In S. Goldwasser, editor, *Proceedings of Innovations in Theoretical Computer Science*, 2012.

[9] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation*, 2011.

[10] A. Igarashi, M. Lackner, O. Nardi, and A. Novaro. Repeated fair allocation of indivisible items. In M. J. Wooldridge, J. G. Dy, and S. Natarajan, editors, *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[11] M. Lackner. Perpetual voting: Fairness in long-term decision making. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[12] J. Li and J. Xue. Egalitarian division under leontief preferences. *Economic Theory*, 54(3):597–622, 2013.

[13] E. Luca, Z. Nagy, and M. Berchez. Water requirements of the main field crops in transylvania (1964–2002). *Journal of Central European Agriculture*, 4(2):97–102, 2003.

[14] D. C. Parkes, A. D. Procaccia, and N. Shah. Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. *ACM Transactions on Economics and Computation (TEAC)*, 3(1):1–22, 2015.